Spoken Dialogue Systems for Information Retrieval with Domain-Independent Dialogue Strategies

Kazunori KOMATANI

Spoken Dialogue Systems for Information Retrieval with Domain-Independent Dialogue Strategies

Kazunori KOMATANI

September 2002

Abstract

One of the vital components for effective human-computer interaction is flexible dialogue management, which understands user's intention and generates appropriate responses. We address dialogue strategies in spoken dialogue systems for information retrieval.

Natural language representation contains ambiguity. The problem becomes serious especially in speech input because speech recognition essentially contains some errors. Ambiguity is also arisen when many candidates are obtained as the query results. Thus, spoken dialogue systems for information retrieval should dissolve these kinds of ambiguity and help users to narrow down the obtained query results by inferring their intention. This is especially important in speech communication because it is not possible to present all query results unlike text or graphical user interfaces.

We address methods to resolve these kinds of ambiguity through dialogue. For the purpose, it is necessary for the system to make confirmation and generate questions. Since redundant confirmation and rigid questions hamper efficient user-friendly interaction, flexible dialogue strategies are indispensable to generate only appropriate responses.

The dialogue strategies should be domain-independent. In simple tasks where all of dialogue procedures can be described manually and the goal of the system can be achieved by system-initiated dialogue, the dialogue strategy including the generation of the confirmation and guidance can be predefined by hand. However, in more complicated and unrestricted tasks such as general information retrieval, static dialogue management based on a domain-dependent specification is not applicable. Therefore, dialogue strategies for information retrieval should be based not on predefined dialogue states described by hand, but solely on either recognition results or information retrieval results.

Based on the standpoint, we focus on the disambiguation in information retrieval with speech interfaces. First, we cope with the recognition errors by making efficient confirmation and guiding utterances using confidence measures of speech recognition results. The other ambiguity caused by natural language expressions and a lot of query

results is resolved by generating guiding questions using the structures of domain knowledge. We also address domain-independent construction of language models and dialogue models. As the basis of domain-independent dialogue models, automatic annotation of the discourse-level tags is studied. Domain-independent construction of spoken dialogue systems is also studied by focusing on the language models. Specifically, we develop a domain-independent platform with a flexible key-phrase spotter based on a combined language model.

In chapter 2, we report the development of a program to infer the utterance-unit tag, which is one of the discourse tags. As it is labor-intensive to annotate large amount of data with discourse-level tags, the automatic tagger is useful. The program utilizes surface features of utterances and the relation of the exchange structure between utterances in order to infer the tags. The experimental result shows the tagging accuracy of 73% in an open test.

In chapter 3, a domain-independent platform of spoken dialogue systems for database query is presented. Conventional development of speech interfaces generally involves much labor cost in either describing a task grammar or collecting a domain corpus. Our platform generates a lexicon and grammars for key-phrase parts by extracting domain database entries and using simple templates. The generated grammar is combined with a word 2-gram model trained with similar domain corpora. Flexible speech understanding is realized by spotting key-phrases based on the combined language model without spoiling portability. We apply this platform to a hotel query system and a literature query system. Experimental evaluation on the generated hotel query system demonstrates that the phrase spotter using the combined language model reduces the interpretation error rate by 15.5% compared with decoding the whole utterance using a fixed grammar.

In chapter 4, we present a method to handle speech recognition errors in a framework of mixed-initiative dialogue, in which the system makes confirmation and guidance using confidence measures (CMs) derived from speech recognition results. We define the word-level confidence measure as a posteriori probability using speech recognition hypotheses and their scores. Using this confidence measure, the system controls generation of confirmation. We also define the confidence measure of semantic categories, which enables effective guidance even when successful interpretation is not obtained. The method is evaluated on the hotel query task using the data collected with the platform, and also on the ATIS and DARPA Communicator tasks. The interpretation error rate is reduced by

8.7% on the hotel query task.

In chapter 5, we address dialogue strategies that narrow down the user's query results obtained by an information retrieval system with speech interfaces. The follow-up dialogue to constrain query results is significant especially with the speech interfaces such as telephones because a lot of query results cannot be presented to the user. The proposed dialogue framework generates guiding questions to eliminate retrieved candidates using the distribution of document statistics and a structure of task knowledge. We first describe its concept on general information retrieval tasks, and propose a hierarchical confirmation strategy by making use of a tree structure of the manual in a query task on an appliance manual where structured task knowledge is available. We formulate three cost functions for selecting optimal question nodes and compare them. Experimental evaluation demonstrates the number of average dialogue turns is reduced by about 30% compared with a baseline method. The result shows that the proposed system helps users find their intended items more efficiently.

Chapter 6 concludes the thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Speech is one of the most familiar media for human communication, so will it be in human-computer communication. Studies on automatic speech recognition get fruitful results in recent years. For example, in the task of Japanese dictation of newspaper articles, speech recognition accuracy over 90% is achieved nearly in real time.

A spoken dialogue system is one of the promising applications of the speech recognition technique. It interprets the speech recognition results to extract necessary information for task completion, and generates a response according to the knowledge source such as a relational database. Such a system enables us to access the enormous data stored in computers without using a special apparatus, and get useful information through the interface natural for us.

So far, many institutes all over the world have developed spoken dialogue systems such as a travel information system and a train timetable query system. Some systems such as the weather information are used practically. However, the tasks of these systems are limited to small and simple ones. The vocabulary size of speech recognition is small (dozen or hundreds of words), and dialogue management and response generation are often so rigid. In such systems, the interaction is organized by the system-initiated dialogue in which the system asks necessary items one by one, and the responses are generated according to the rules described manually. So, users are allowed to utter only what the system requires. Otherwise, the dialogue often fails because of speech recognition errors.

To realize user-friendly and flexible interaction, it is desirable that users can express their intention to the system in their own manners and with their initiatives. The most significant problem is ambiguity caused by speech recognition and language understanding by allowing more freedom to users. Flexible dialogue management is highly required in

order to cope with the ambiguity.

Natural language representation inherently contains ambiguity. The problem becomes serious especially in speech recognition because it essentially contains errors. Ambiguity is also arisen when many candidates are obtained as the query results. Thus, spoken dialogue systems for information retrieval should dissolve these kinds of ambiguity and help users to narrow down the obtained query results by inferring their intention. This is especially important in speech communication because it is not possible to present all query results without text or graphical user interfaces.

These kinds of ambiguity should be resolved through dialogue. For the purpose, it is necessary for the systems to make confirmation and generate questions. However, redundant confirmation and rigid questions hamper efficient user-friendly interaction. Thus, flexible dialogue strategies are indispensable to perform only appropriate responses.

The dialogue strategy should be domain-independent. In simple tasks where all of dialogue procedures can be described manually and the goal of the system can be achieved by system-initiated dialogue, the dialogue strategy including the generation of the confirmation and guidance can be specified beforehand by hand. However, in more complicated and unrestricted tasks such as information retrieval, such methodology is not applicable. Therefore, dialogue strategies in spoken dialogue systems for information retrieval should be based not on predefined dialogue states described by hand, but on either recognition results or information retrieval results.

In spoken dialogue systems, a domain-independent methodology of configuration is also desirable. The conventional systems usually incorporate the domain-specific knowledge sources on vocabulary and grammars, which lacks for generality. Such a system often faces problems with the change of the vocabulary and knowledge in the target domain. Moreover, the components of the system cannot be ported to other domains and a totally new system must be designed for a different domain. Thus, the portable modeling of spoken dialogue systems is also significant.

In the thesis, we address spoken dialogue systems for information retrieval with flexible dialogue strategies. The dialogue strategies and the language modeling are pursued in a domain-independent manner. The methodologies are expected to be applied to various domains.

1.1 Problems

Portability in Language Models and Dialogue Models

First, we address the domain-independent construction of language models and dialogue models. As the basis of domain-independent dialogue models, corpora annotated by discourse-level tags are needed. The language model covering vocabulary and expressions specific to the domain is also indispensable to perform flexible recognition and understanding.

It is very labor-intensive to construct the annotated corpus by hand. A large amount of annotated corpora is needed as a basis of statistical methods, which have succeeded in various fields such as speech recognition. Usually, to construct a large amount of annotated corpora, programs such as morphological analyzers and syntactic analyzers are utilized. However, there is neither established methods nor a program that performs discourse-level analysis and helps annotation of the discourse tags.

Another problem is construction of language models covering domain-specific vocabulary and expressions. In conventional methods, grammar rules written by hand have been used as domain-specific language models. The grammar-based model can easily introduce domain knowledge. However, it can hardly attain sufficient performance in recognizing spontaneous speech, because it is difficult to cover various utterance patterns. On the other hand, statistical language models trained with large corpora are more suitable to recognize user utterances flexibly. However, it is also difficult to collect training corpora sufficiently corresponding to every target domain. Thus, portable modeling that can accept various utterances flexibly and reflect the domain-specific knowledge explicitly is highly needed.

Handling Speech Recognition Errors

The most critical cause of the ambiguity in information retrieval systems with the speech interfaces is speech recognition errors, since recognition errors are essentially inevitable unlike text or graphical user interfaces. Thus, spoken dialogue systems must be robust enough to complete the task by handling recognition errors.

In the past works on dialogue systems, several complicated models such as plan recognition from user's utterances [1, 2], inference of user model and generation of cooperative answers [3, 4] are elaborated. Most of the conventional studies using such complicated

processes assumed that the input to the system contains no error. Even when the existence of recognition errors is assumed, the error is approximated as replacement of some words. This assumption is too simple because the recognition errors often happen consecutively in actual speech recognition.

In conventional dialogue systems using speech recognition, confirmation is generated based on domain-dependent specification of the dialogue flow. Originally, confirmation to handle speech recognition errors should be generated according to each recognition result. Fixed confirmation procedure makes the dialogue redundant. Therefore, confirmation strategies should be flexible based not on predefined domain-dependent descriptions but on the situations of the individual recognition.

Handling Ambiguity in Query Results

The other ambiguity is caused by natural language expressions and a lot of query results. In the communication between a user and the system, user intention is mapped into natural language expressions as an utterance, and then it is matched with the system knowledge. The system generates outputs according to the matching results. Ambiguity is included in the both processes: one is arisen between user intention and natural language expression, and the other is during the matching with the system knowledge. It is increased if there are speech recognition errors. Because of the ambiguity, many candidates are obtained as the retrieval results than the user originally intended. The ambiguity should be dissolved through interactions such as appropriate guiding questions. Especially for novice users, the guidance is significant.

In conventional systems, such guidance is generated based on rules described by hand. However, in more complicated tasks such as information retrieval, all of dialogue procedures cannot be described manually. The guidance should be generated based not on the static description predefined manually, but on the general procedure using either speech recognition results or information retrieval results.

In former works, knowledge sources of the target domain in spoken dialogue systems are generally represented as a well-organized format such as relational database (RDB). But recently, document retrieval is performed not only for such well-organized data but also for documents described in natural language and WWW. Thus, it is desirable that the guidance strategy does not assume the well-organized format.

1.2 Domain-Independent Approach

To solve the above problems, we propose domain-independent approaches in dialogue management.

1.2.1 Domain-Independent Generation of Language Model and Annotation of Discourse Tag

We model that the utterances consist of key-phrases and filler phrases. The filler or carrier phrases are of variety, but are not domain-specific. So, we apply statistical language models trained with similar domain corpora to the filler phrases. The key-phrases depend on the target domain, but its representation is definite. So, we derive grammar rules for this part from the domain database and template patterns. Thus, we generate the combined language model by integrating the grammar rules and statistical language model, which realizes both flexibility and portability.

Annotation of discourse tags should also be achieved in a domain-independent manner. We develop a program to infer the utterance-unit tag, which is one of the discourse tags representing most common communicative aspects of utterances. In inferring the tag, we focus on surface features of utterances and the relation of the exchange structure between utterances. These features appear generally in task-oriented dialogues, and therefore are not dependent on the specific domain.

1.2.2 Confirmation Strategy using Two-level Confidence Measures of Speech Recognition

We cope with recognition errors through dialogue by generating efficient confirmation and guidance. The confirmation strategies have been deliberated in conventional studies [5, 6, 7, 8]. In these studies, however, speech recognition errors are simulated with one fixed error rate throughout the dialogue, which is not the case in actual automatic speech recognition. In practical situations, dialogue systems should adapt its dialogue strategy dynamically according to the degree of recognition errors. Namely, it should generate appropriate questions and guidance depending on the situation.

In order to generate confirmation efficiently, one problem is how to distinguish between correct hypotheses and recognition errors. It is necessary to define a criterion indicating whether the candidate is erroneous or not. With the criterion, the system can make

confirmation only for the uncertain part that is probably erroneous. Another problem is to set a threshold to the criterion that decides the candidate should be accepted, confirmed or rejected. The threshold should be set optimally not to accept incorrect candidates and not to generate redundant confirmation. Furthermore, when the recognition does not go well, it is desirable that the system presents acceptable patterns to the user, which will guide the user's utterances into the system's capability and prevent further recognition errors.

We define two-level confidence measures: one for content words and the other for semantic attributes. Word-level confidence measure is used to decide whether to generate confirmation. The threshold for the word-level confidence measure is determined optimally considering the balance between acceptance and rejection. Furthermore, the other confidence measure for semantic categories enables the system to generate effective guidance even when any confident interpretation cannot be obtained on the content-word level. Then, mixed-initiative dialogue, in which the user can make utterances spontaneously and the system takes the initiative if necessary, is realized.

The confirmation strategy utilizes only recognition results and semantic category structures, which do not contain domain-specific knowledge. Therefore, flexible interaction to manage speech recognition errors is realized in a domain-independent manner.

1.2.3 Guidance Strategy using Structure of Domain Knowledge

We address dialogue strategy to dissolve ambiguity caused by queries with natural language and a lot of query results. We propose a framework to generate guiding utterances in tasks where knowledge sources are represented in less rigid format, which means the slot-type semantic structure is not assumed. The strategy utilizes the distribution of document statistics and the structure of task knowledge, which is formulated in a domain-independent manner.

We first discuss dialogue management in a task where domain knowledge is represented as a set of keywords for data entries. This assumption stands for general document retrieval. We also describe dialogue management in a task where concept hierarchy in the domain is available. The hierarchy is typically obtained as a tree structure of a table of contents of the documents. For both cases, the optimal question to guide users is selected. The selection is done by defining heuristic cost functions, which take account of either an information theoretic criterion or the expected number of following questions.

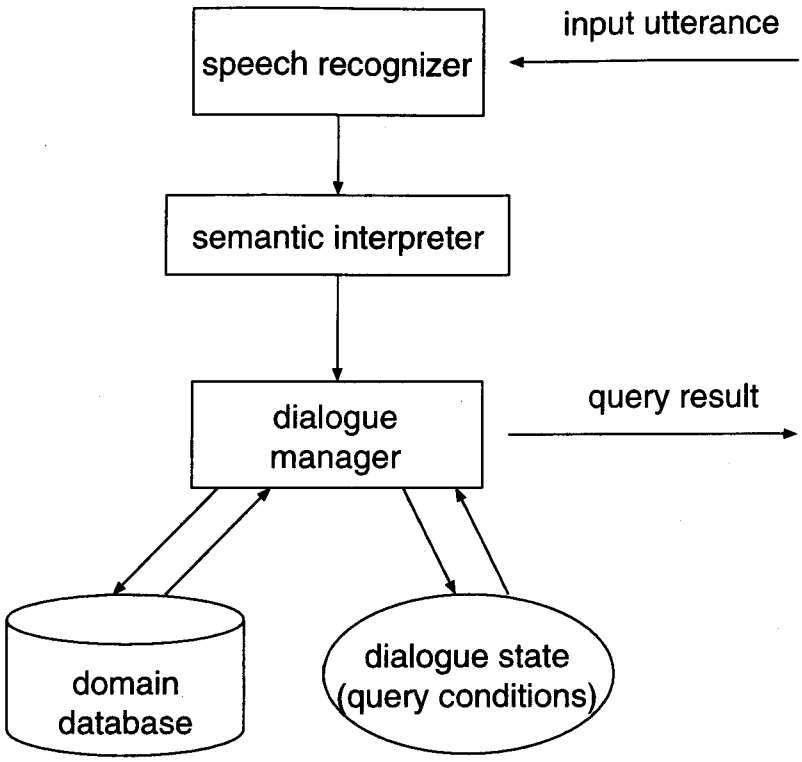


Figure 1.1: Flowchart of spoken dialogue system

The generated guidance helps users narrow down retrieved results most efficiently toward their intended items.

1.3 Basic Architecture of Spoken Dialogue System

Spoken dialogue systems generally consist of the following components as shown in Figure 1.1.

1. speech recognizer
2. semantic interpreter
3. dialogue manager

Speech Recognizer

A speech recognizer accepts a user utterance, and transforms it to a word sequence. Speech recognition is a process to generate the most likely word sequence W for a given

speech input X.

$$W = \arg\max_{w} P(w|X) \tag{1.1}$$

This probability can be written as follows based on the Bayes' rule.

$$P(w|X) = \frac{P(X|w)P(w)}{P(X)} \tag{1.2}$$

As $P(X)$ does not affect the selection of w, speech recognition process is formulated as follows.

$$\arg\max_{w} P(w|X) = \arg\max_{w} P(X|w)P(w) \tag{1.3}$$

$P(X|w)$ means the output probability of speech X for a word w, which is given by an acoustic model. $P(w)$ means the probability how plausible a word w appears in the context. It is given by a language model. Speech recognition is to search for a word sequence where the product of these two probabilities gives the maximum.

For spoken dialogue systems, we can use general acoustic models as long as they are trained in similar environments, that is, distribution of speakers and input channel conditions. On the other hand, the language model must be adapted to each domain because a general language model does not contain vocabulary and expressions specific to the domain such as proper nouns. This issue is addressed in chapter 3.

Semantic Interpreter

Semantic interpreter receives a word sequence from the speech recognizer, and transform it to a format necessary for task completion. In a database query task, content words and corresponding search items are extracted from the recognized word sequence, and are transformed to the query keys.

In conventional spoken dialogue systems, the speech recognizer conveys the results to the semantic interpreter, and the interpretation results are also passed to the dialogue manager, sequentially. In general, as a speech recognition program and a natural language processing program are developed independently, the above composition is derived straightforward. In such a case, it is possible that a semantic interpreter cannot accept the recognized results. Another approach is a key-phrase spotting, which does not decode the whole utterances but focuses on particular phrases. In the key-phrase spotting method, interpretation mechanism is simple and always output some results. The method is also robust against variation of utterances and recognition errors. Therefore, we adopt the strategy.

Dialogue Manager

A dialogue manager executes the following processes based on the results from the semantic interpreter.

- Management of confirmation and guidance
- Execution of query (interface between the domain database)
- Generation of responses
- Management and update of dialogue states

A conventional dialogue model based on an automaton model changes its dialogue state according to the input. This model, which is adopted in Voice XML, is suitable to describe simple tasks where the system-initiative dialogue is applicable, because the automaton model can be designed intuitively. However, it requires a great deal of labor in the description in complicated tasks and is not applicable to the information retrieval task.

We propose a dialogue management not by describing the behaviors for all states by hand, but in a domain-independent manner. As shown in Figure 1.2, confirmation and guiding questions to manage speech recognition errors are generated by using the semantic category structure and two-level confidence measures, which are derived from N-best candidates of speech recognition. Guidance to constrain obtained query results is generated by selecting the optimal questions by using the structure of domain knowledge and current query conditions. The two strategies realize efficient confirmation and guidance based on the domain-independent information such as the structure of domain knowledge, the structure of semantic category, speech recognition results and current query conditions. Therefore, flexible dialogue management is realized without describing any domain-specific rules for every target domain.

1.4 Tasks in Spoken Dialogue Systems

The tasks for spoken dialogue systems can be classified into three types according to the direction where information flows [9] as shown in Table 1.1. The information retrieval task that is the target in this thesis is included in the database query class in Table 1.1.

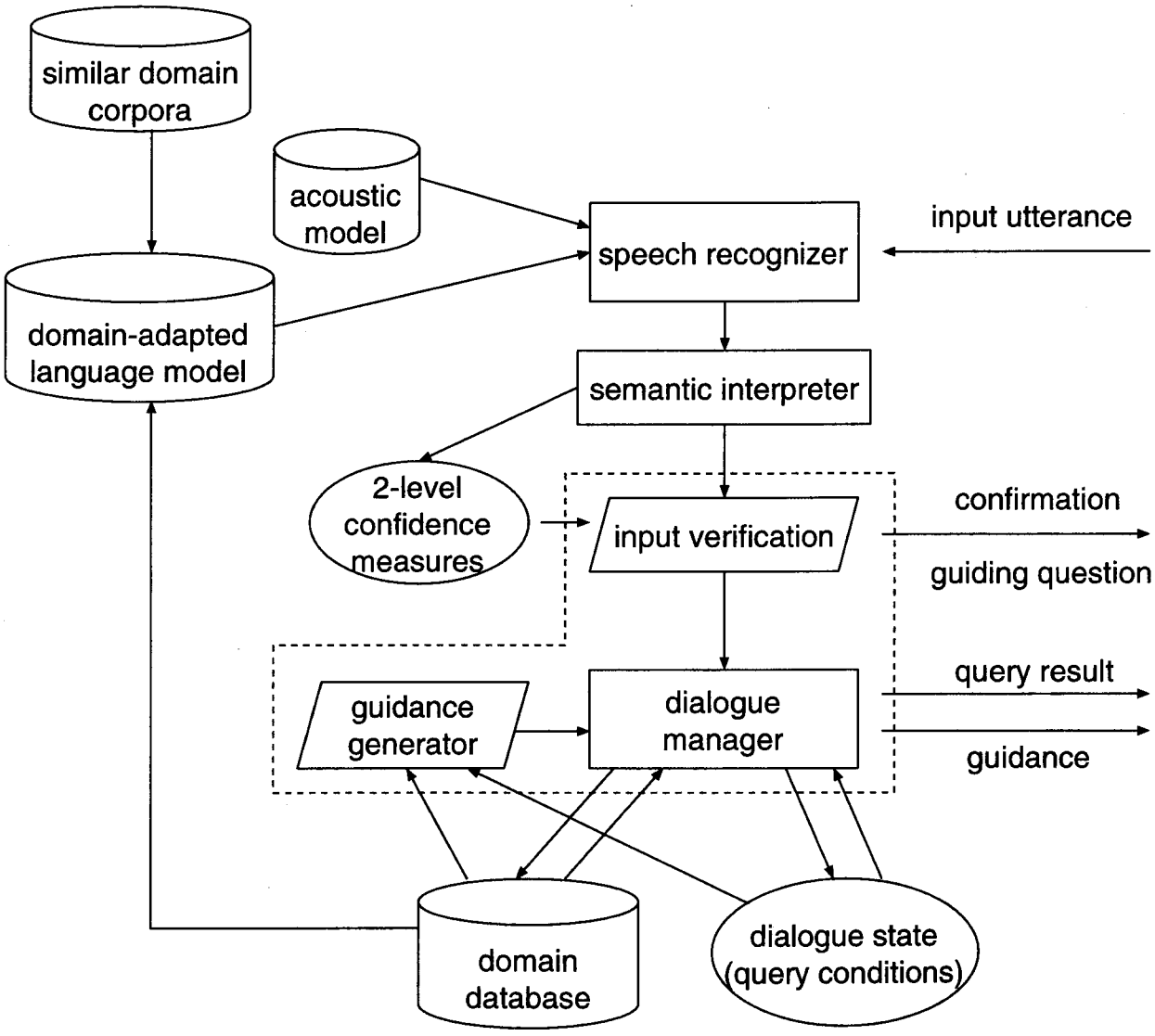


Figure 1.2: Overview of the proposed spoken dialogue system

Table 1.1: Class of abstract tasks

information flow	abstract task	example domain
user $\rightarrow$ system	slot-filling	telephone shopping on-line trade
user $\leftrightarrow$ system	database query	book order bibliography search
user $\leftarrow$ system	explanation	route direction instruction manual

In a slot-filling task, information is transmitted from a user to the system in a single direction. For such tasks, dialogue management can be realized simply by a system-initiated strategy because the dialogue procedure is fixed. The task is accomplished by asking the required information in turn, for example, a telephone number and a code number of commodities in a telephone-shopping task.

On the other hand, we deal with tasks in the database query class, in which information is transmitted in both directions, where a user receives the results of database query from the system, and then constrain or relax the condition based on the obtained results. It is impossible to describe the dialogue procedure statically because the task is not accomplished by filling all predefined database items unlike tasks in the slot-filling class. The hotel query task, which is one of the target tasks in the thesis, belongs to the database query class. In these tasks, user utterances are modeled as setting and retracting search keys according to the previous results dynamically. So, the dialogue flow is variable depending on the retrieved results obtained by the specified query condition. Namely, the goal in the database query class is to narrow down the search results into a few items, which satisfy the user's intention. This goal cannot be realized by asking the contents of slots in a fixed order, since all the slots do not necessarily have to be filled. Therefore, we propose a strategy to generate system's utterances using the distribution of document statistics and the structure of task knowledge.

1.5 Outline of the Thesis

This thesis is organized by the following chapters.

In chapter 2, we review the current studies on discourse tags, and present a method to annotate an utterance-unit tag automatically, which is one of the discourse tags. As it is

labor-intensive to annotate large amount of data with discourse-level tags, the automatic tagger is useful. The program utilizes surface features of utterances and the relation of the exchange structure between utterances in order to infer the tags. The proposed method is experimentally evaluated on spoken dialogue corpora.

In chapter 3, we describe a domain-independent platform of spoken dialogue systems for database query, and discuss how to compose the language models. Conventional development of speech interfaces generally involves much labor cost in either describing a task grammar or collecting a domain corpus. Our platform generates a lexicon and grammars for key-phrase parts by extracting domain database entries and using simple templates. We propose a flexible key-phrase spotting method using the combined language model, which is composed by integrating grammar rules reflecting domain-specific knowledge and a statistical model covering various expressions. Flexible speech understanding is realized by spotting key-phrases based on the combined language model without spoiling portability. We apply this platform to a hotel query system and a literature query system.

In chapter 4, we present a dialogue strategy using confidence measures of speech recognition results. We define the word-level confidence measure as a posteriori probability using speech recognition hypotheses and their scores. Using this confidence measure, the system controls generation of confirmation. We also define the confidence measure of semantic categories, which enables effective guidance even when successful interpretation is not obtained. Furthermore, we describe a method to determine optimal thresholds for the dialogue strategy by defining loss functions to take the balance between acceptance and rejection. Then, mixed-initiative dialogue, in which the user can make utterances spontaneously and the system takes the initiative if necessary, is realized.

In chapter 5, we address dialogue strategies that narrow down the user's query results obtained by an information retrieval system with speech interfaces. The follow-up dialogue to constrain query results is significant especially with the speech interfaces such as telephones because a lot of query results cannot be presented to the user. The proposed dialogue framework generates guiding questions to eliminate retrieved candidates using the distribution of document statistics and a structure of task knowledge. We first describe its concept on general information retrieval tasks, and propose a hierarchical confirmation strategy by making use of a tree structure of the manual in a query task on an appliance manual where structured task knowledge is available. We formulate three cost functions for selecting optimal question nodes and compare them experimentally.

The generated guidance helps users narrow down retrieved results most efficiently toward their intended items.

Chapter 6 concludes this thesis.

Chapter 2

Automatic Annotation of Discourse Tags in Dialogue Corpora

2.1 Introduction

Recently, the importance of corpora is getting greater owing to the success of statistical methods in speech and natural language processing. It is labor intensive to annotate large amount of data as well as to collect and transcribe them. Automatic taggers are highly needed to reduce the cost of annotation. Actually, for annotation of morphological and syntactic-structure information, morphological analyzers (part of speech taggers) and syntactic parsers are utilized, respectively. The use of such tools helps to reduce the cost of annotation. For a higher level of semantic or discourse information, there is not an automated analyzer that realizes sufficient accuracy. Therefore, annotation of discourse-level tags is done manually and costs a great deal.

We propose a method to infer the utterance-unit tag, which is one of the discourse-level tags to spoken dialogue corpora. We assume only surface text and the results of morphological analysis (part of speech information) as input data in order to avoid problems that are characteristic in spoken dialogue such as ellipsis and a repair utterance. As existing syntactic parsers cannot be used for written language, they cause errors for spoken dialogue corpora. We also develop a GUI tool that helps discourse-level tagging. As one of the functions of this GUI tool, we implement the proposed method to infer the utterance-unit tag, and evaluate the method by experiments.

2.2 Review of Discourse Tagging

2.2.1 Standardization of Discourse Tags

Since reliable corpus annotation is difficult and labor-intensive especially in higher-level tagging, it is desirable to design a common tag set and to share the annotated corpora. The Discourse Research Initiative (DRI) was set up in March 1996 by American, European and Japanese researchers to develop standard discourse annotation schemes [10, 11]. In accordance with the effort of this initiative, the discourse tagging working group has started in Japan in May 1996 under the support of the Japanese Society of Artificial Intelligence. The working group has engaged in standardization of Japanese discourse annotation schemes [12]. The standardization and systematization of the Japanese discourse annotation scheme are proposed for utterance units, discourse structure and discourse markers [13]. In this chapter, we deal with the utterance-unit tag.

The utterance-unit tag comes from speech act theory [14]. By the speech act theory, three acts are performed simultaneously with every utterance a person makes. The three acts, which are advocated by J. L. Austin, are described as follows [15].

1. Locutionary act
   Something is *said.*

2. Illocutionary act
   A *communicative function* (inviting, apologizing, requesting) is *performed.*

3. Perlocutionary act
   An *effect is produced* on the listener or hearer.

Among the three acts, appropriate information for the annotation unit should satisfy the following two conditions.

- A contribution to solve the problem in dialogues is large.
- An objective description is possible.

The locutionary act is expressed by a transcribed sentence itself. The perlocutionary act is difficult to describe objectively because a guess for mental states of the interlocutor is needed. Therefore, it is appropriate to annotate the utterance units by information

representing the illocutionary act, which can contribute a problem solving and can be described objectively.

The utterance-unit tag mentioned above represents the illocutionary act. It cannot be determined by only surface information of utterances essentially, and should be classified by the functional aspect of the utterances.

2.2.2 Utterance Units for Annotation

Logical units such as sentences or utterances are ordinarily presumed to annotate discourse-level tags. The transcription of speech in a dialogue corpus is usually divided by either definite changes of speakers or pauses longer than some threshold. But this physical unit does not necessarily agree with the logical unit representing the meaning. Especially in conversational situations, turn takings by "*aiduti* (back-channel feedback)", which does not constitute logical units, occur frequently as modeled in [16]. Therefore, it is necessary to reorganize (divide or combine) text corpora into logical units suitable for tagging as the first step of discourse-level annotation. We call the unit as "annotation unit" hereafter. Also in conversational situations, transcribed portions that are not recognized by the interlocutor, such as *aiduti*, misstatements, fillers, disfluency and mutters, cannot be the logical unit, and therefore need to be marked such.

However, there are many opinions about how to determine the annotation unit. In [17], a method to divide one utterance using linguistic features and prosodic ones is proposed. Meteer proposed an annotation unit named as "slash unit" that defines the handling of incomplete sentences, ungrammatical sentence elements and responses [18]. In [18], a complex sentence is defined such that a subordinate clause is regarded as being in the same unit with a main clause, and that a complex sentence by a coordinate conjunction is divided into plural units by judging from the existence of a subject.

Thus, the annotation unit cannot be defined uniquely. Moreover, there is a dependency between annotation units and utterance-unit tags mutually. That is, while an annotation unit is settled as an object of tagging, tagging cannot be done without setting annotation units. Considering these analyses, we assume that the annotation units should be divided or combined using the other information sources such as prosody and the definition of the slash unit. We assume that the input has already been divided or combined into appropriate annotation units beforehand.

Here, we show an example of a transcribed dialogue divided by either definite changes

U: '*Kouenkai wo*' <416>
(A lecture meeting ..)

S: '*Hai.*' <224>
(Uh.)

U: '*Getsuyou no*' <336> '*Niji kara*' <464>
(From two o'clock on Monday,)

U: '*Goji made touroku shite kudasai*' <976>
(Register to five o'clock.)

S: '*Hai.*' <176> '*Kouenkai wo getsuyou no*' <320> '*Juuyoji kara juushichiji made touroku shimasu.*' <304> '*Yoroshii desu ka?*' <448>
(Yes. I am going to register a lecture meeting from fourteen to seventeen. Is this correct?)

U: '*Hai.*' <1152>
(Yes.)

Numbers in <> mean length of pauses ([ms])

Figure 2.1: Example of a transcribed dialogue

of speakers or pauses longer than a threshold (400 milliseconds here), and an example of its reorganization into the annotation units. The transcription in Figure 2.1 is divided into annotation units shown in Figure 2.2. Thus, it is obvious that a physical pause does not always agree with the annotation unit.

2.2.3 Annotation Scheme

We present the set of utterance-unit tags. The tagging scheme we address in this chapter is based on the discourse tagging working group in Japanese Society of Artificial Intelligence [12]. In the tagging scheme, a structure of task-oriented dialogue, which is a target of the tagging, is constituted like Figure 2.3. But this constitution is not regarded as so definite. The utterance-unit tags are defined by sub-dividing each portion of ⟨initiate⟩ ⟨response⟩ ⟨follow-up⟩ ⟨response/initiate⟩ as shown in Figure 2.4.

U: *Kouenkai wo {Hai.} Getsuyou no niji kara goji made touroku shite kudasai.*
(Please register a lecture meeting from two o'clock to five o'clock on Monday.)

S: *Hai.*
(Yes.)

S: *Kouenkai wo Getsuyou no juuyoji kara juushichiji made touroku shimasu.*
(I am going to register a lecture meeting from fourteen to seventeen.)

S: *Yoroshii desu ka?*
(Is this correct?)

U: *Hai.*
(Yes.)

The {} means *aiduti* (back-channel feedback)

Figure 2.2: Annotation units corresponding to the transcript

2.2.4 A Tool to Support Annotation of Discourse Tagging

We implement "jdat (Japanese Dialogue Annotate Tool)" that supports the annotation visually by mouse operation. In order to construct large corpora annotated by discourse tags, tools that support tagging are required. In the annotation of the utterance-unit tag, it is required to reorganize transcribed text into the annotation units, and to specify fillers, disfluency and *aiduti*. To support such works reduces the labor in the annotation.

Annotators can easily specify an utterance-unit tag by selecting from displayed items. An outlook of the tool is shown in Figure 2.6. The jdat is made by porting the existing discourse-tagging tool (dat [19]) used in Europe and America into Japanese, and adding some functions. It can provide reference information such as a speech sound and results of syntactic analysis, which help human annotators to decide the tags as shown in Figure 2.5. The proposed method to infer utterance-unit tags is implemented as one of functions of the jdat.

Overall structure

task-oriented dialogue → ⟨dialogue management [Open]⟩*
⟨exchange structure⟩+ ⟨dialogue management [Close]⟩*

Basic pattern

⟨exchange structure⟩ → ⟨initiate⟩ ⟨response⟩ (⟨follow-up⟩) (⟨follow-up⟩)

Embedded pattern

⟨exchange structure⟩ → ⟨initiate⟩ ⟨embedded structure⟩* ⟨response⟩
(⟨follow-up⟩) (⟨follow-up⟩)

⟨embedded structure⟩ → ⟨response/initiate⟩ ⟨response⟩

- The '*' and '+' denote the repetition over 0 time and over 1 time, respectively.
- Symbols in () can be omitted.

Figure 2.3: Patterns of exchange structure

Dialogue Management:
Open, Close

Initiate:
Request, Suggest, Persuade, Propose, Confirm, Yes-No question, Wh-question, Promise, Demand, Inform, Other assert, Other initiate

Response:
Positive, Negative, Answer, Hold, Other response

Follow up:
Understand

Response with Initiate:
The element of this category is represented as *Response Type / Initiate Type.*

Figure 2.4: Tag set of the utterance-unit tag

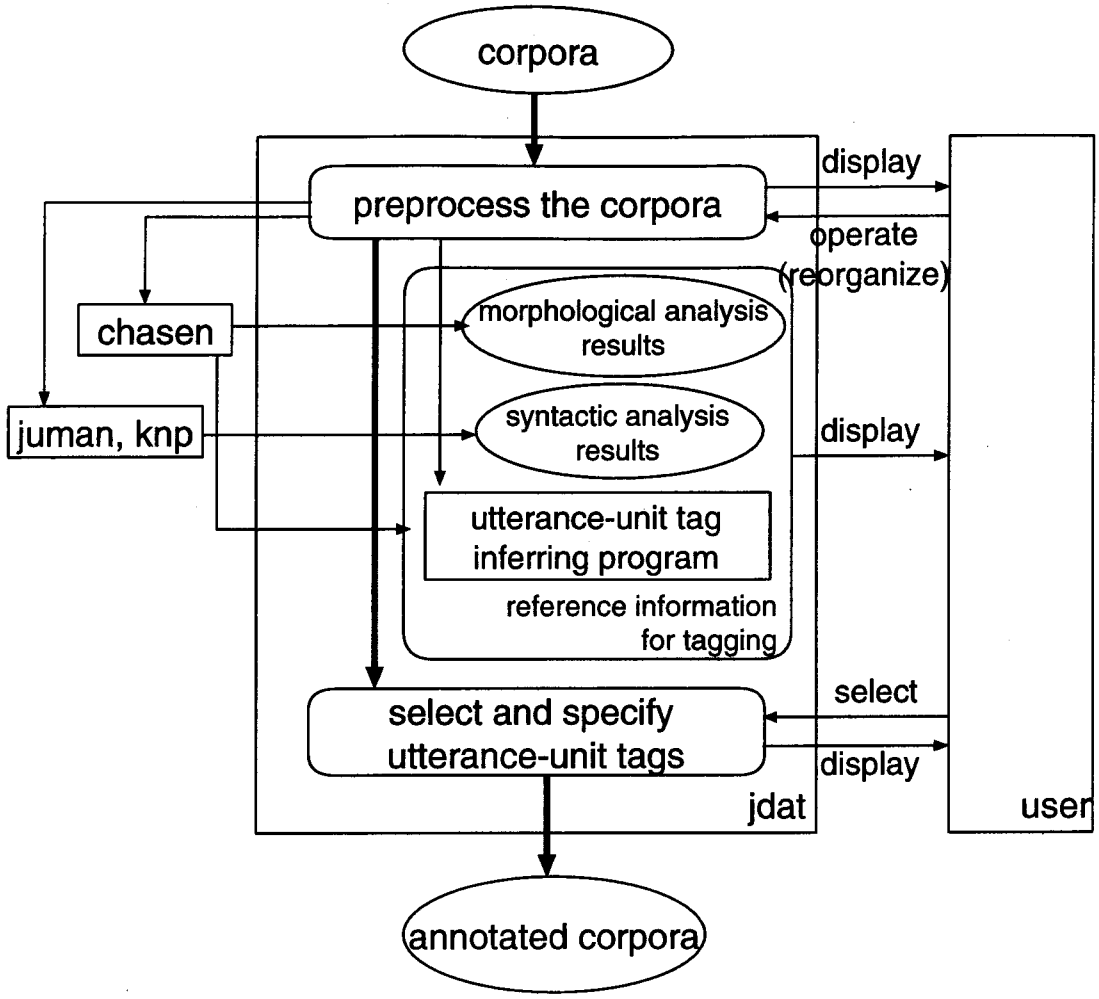


Figure 2.5: Data flow in the annotation tool (jdat)

2.3 Method to Infer Utterance-Unit Tag

2.3.1 Related Works

Some methods to infer the illocutionary act have been studied until now. It is important in dialogue processing to infer an intention and a purpose in utterances of the interlocutor. The intention in utterances is modeled as an illocutionary act. In [20], Takinaga et al. proposed a method that extracted an intention, a topic and a focus from a user's utterance by setting the task of a sightseeing guidance. This method performs the inference using domain-dependent knowledge that was constructed manually by grouping all the words semantically and making the topic templates of the task. However, such an inferring method using domain-dependent knowledge written by hand cannot be applied in constructing large-scale corpora annotated with discourse tags in various tasks.

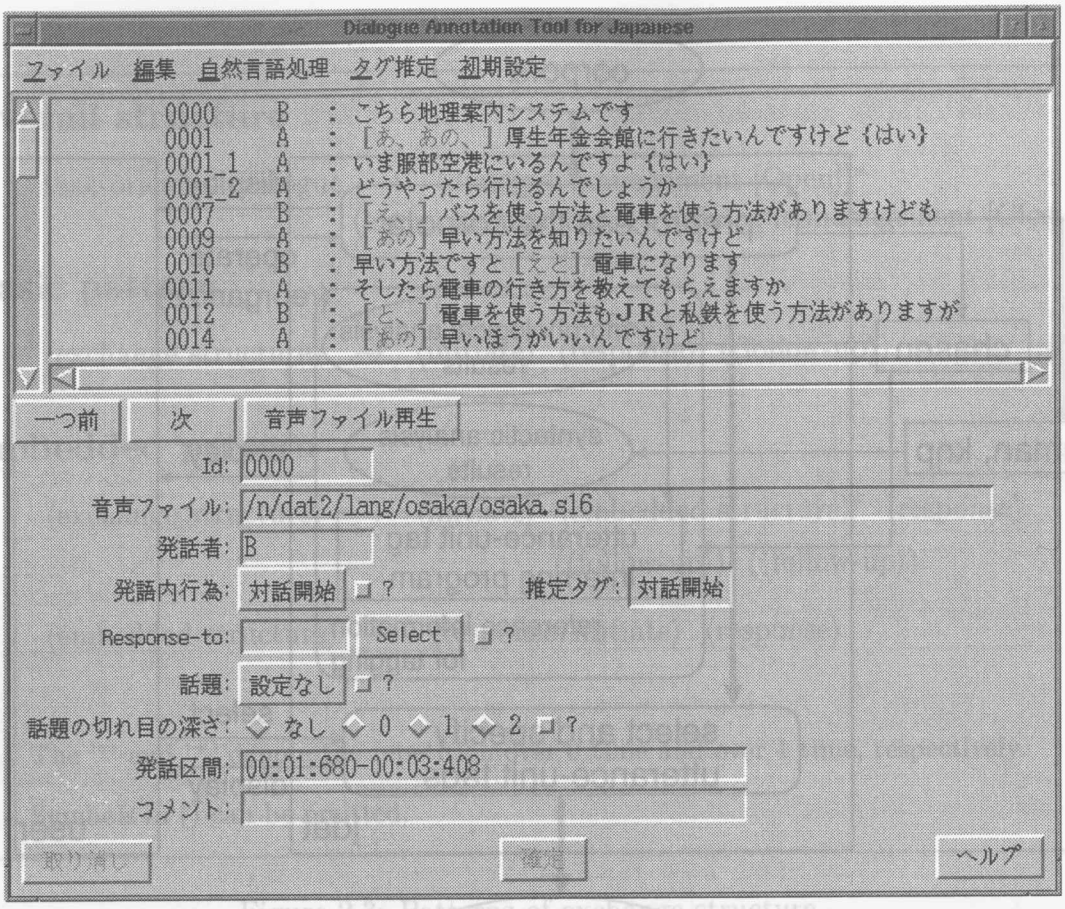


Figure 2.6: Outlook of the annotation tool (jdat)

Moreover in [21], dialogue acts were predicted in order to select keywords in the following user's utterance in a machine translation system. Because our method is not needed to predict the next utterances online but can assume already transcribed spoken corpora as the input, features in the succeeding utterances can be used as a key for the inference.

Based on the analysis, our method infers the illocutionary act without using domain-dependent knowledge as an input. We focus on the characteristics of the surface expression appearing in the "initiate-response" units in task-oriented dialogues. We call this unit "exchange unit" hereafter. The exchange units are inferred first, and then utterance-unit tags are decided as a subdivision of the exchange unit. By inferring the exchange unit beforehand, correspondence between an "Initiate" part and a "Response" part can be identified even when the exchange unit is nested and the two parts are separated. The correspondence is important to determine the tags in the response portion.

2.3.2 Input to the Inference Program

We assume the following three items as an input to the inferring program.

1. Transcription

2. Morphological analysis results of the transcribed corpora

3. Specification of ⟨dialogue management⟩ parts ([*Open*] and [*Close*])

First, the transcribed corpora must be reorganized (divided and combined into utterance units) as discussed in section 2.2.2.

As well as using the morphological analysis results, we considered to use a syntactic parser's output to infer the utterance-unit tag. However, spoken dialogue corpora contain the various problems characteristic to spoken language, such as ellipses of a predicate or a post-positional particle in Japanese. As it is difficult to define a sentence structure systematically for imperfect sentences including such ellipses, existing syntactic parsers developed for written language will output a different structure if a correct predicate is complemented. So, we have not adopted the output of syntactic parsers. We show the examples in which different outputs are obtained in Figure 2.7 and Figure 2.8.

On the other hand with morphological analyzers, a small gap of the notation or abbreviation of syllables can be solved by registering corresponding words on the dictionary. Then, we use only the morphological analysis result whose accuracy is improved by registering surface expressions, which may cause errors because of characteristics of spoken language.

As for ⟨dialogue management⟩, we assume that [*Open*] and [*Close*] in this part have already been specified manually by annotators as one of the input information. In this part in spoken dialogue corpora, there are various idiomatic expressions. They are highly dependent on the domain and therefore do not necessarily contain domain-independent characteristics. Also, these parts can easily be specified manually because they are at the beginning or the end of dialogue, and the number of these parts is small. Moreover, because the part annotated as [*Open*] is at the beginning of the dialogue, an incorrect assignment of the tag tends to affect the succeeding parts harmfully.

Kochira no ---+
hou ha ---+
suuryou ha.

'*Kochira no hou ha suuryou ha?*'
(How many for this one?)

Figure 2.7: Sentence structure with ellipsis of a predicate

Kochira no ---+
hou ha ---+
suuryou ha ---+
ikutsu desu ka.

'*Kochira no hou ha suuryou ha ikutsu desu ka?*'
(How many do you want for this one?)

Figure 2.8: Sentence structure when a predicate is complemented

2.3.3 Inference of Exchange Units

We explain inference of the exchange unit in this section. We assume that an interaction in a task-oriented dialogue consists of the partial structures: "initiate" and the corresponding "response". Here, we call the partial structure as "exchange unit". As this exchange unit is used in order to infer the partial structure roughly, it slightly differs from the "exchange structure" shown in section 2.2.3 for defining utterance-unit tags.

We assume that the exchange unit consists of the following components.

$$\langle\text{exchange unit}\rangle \rightarrow \langle\text{initiate}\rangle\ \langle\text{response}\rangle\ (\langle\text{follow-up}\rangle)\ (\langle\text{follow-up}\rangle)$$

() denotes that it can be omitted. Moreover, a new exchange unit can be inserted after the ⟨initiate⟩ part as follows. '+' denotes the repetition over 1 time.

$$\langle\text{exchange unit}\rangle \rightarrow \langle\text{initiate}\rangle\ \langle\text{exchange unit}\rangle^{+}\ (\langle\text{response}\rangle)\ (\langle\text{follow-up}\rangle)\ (\langle\text{follow-up}\rangle)$$

#11 S: *'Heya ha dou shimasyou ka?'*
(How about the room?)

#12 U: *'Heya no aki guai ha dou desyou ka?'*
(Are there any vacant rooms?)

#13 S: *'Getsuyou no juuyoji kara juurokuji made syoukaigishitsu ga shiyou kanou desu.'*
(The small conference room is available from fourteen to sixteen on Monday.)

#14 U: *'Deha, syoukaigishitsu de onegai shimasu.'*
(Then, please reserve the small conference room.)

#15 S: *'Wakarimashita.'*
(All right.)

Figure 2.9: Sample dialogue for explaining exchange units

When a new exchange unit is inserted after the ⟨initiate⟩ part, the ⟨response⟩ part may be omitted because the question in the first ⟨initiate⟩ part may be answered in the inserted exchange unit. We show an example of the exchange unit for the dialogue in Figure 2.9. In the sample dialogue in Figure 2.9, the ⟨initiate⟩ utterance (#12) is inserted after the ⟨initiate⟩ utterance (#11). So, if we denote an exchange unit by brackets (), the exchange unit in the sample dialogue is represented as follows.

(#11 (#12 #13) #14 #15)

Utterance #13 is the ⟨response⟩ to utterance #12, and utterance #14 is the ⟨response⟩ to utterance #11. Utterance #15 is the ⟨follow-up⟩. When the utterance having the same function is succeeded by the same speaker or the same utterance is repeated by a request like "Please say once again.", exchange units for those parts are unified.

We take notice of the feature in the ⟨initiate⟩ utterances in order to infer the exchange unit. We assume that the ⟨initiate⟩ utterances in the task-oriented dialogue can be classified into the following two items.

(1) Requesting a certain act or information for the interlocutor

(2) Providing certain information for the interlocutor

- [decisional particle] + [sentence-final particle (interrogative)]
- [auxiliary verb ("*Darou*" type)] + [sentence-final particle (interrogative)]
- [adjective ("*Na*"-inflectional type)] + [sentence-final particle (interrogative)]
- [verb-type suffix] + [sentence-final particle (interrogative)]
- [verb-type suffix] + [auxiliary verb (negative)] + [sentence-final particle (interrogative)]
- the sentence end is imperative base
- "... *shitai n desu kedo (keredomo, kedomo, keredo)*"
- "... *ga ii n desu kedo*"
- "*onegai shimasu (onegai itashimasu)*"
- "... *ha*"

[sentence-final particle (interrogative)] consists of "*ka*" and "*kke*".

Figure 2.10: Features to extract the ⟨initiate⟩ part

"*Hai.*" "*Wakarimashita.*" "*Soudesu.*" "*Soudesuka.*"

Figure 2.11: Expressions that only accept information

First, we assign the ⟨initiate⟩ to the utterances having the features shown in Figure 2.10. The features appear at the end of sentences having function (1). We use the organization of a part of speech in the Japanese morphological analyzer ChaSen [22].

Next, utterances succeeded by the expressions shown in Figure 2.11 are assigned as the ⟨initiate⟩ part corresponding to (2), except for the utterances already assigned by (1). This is because the ⟨response⟩ part corresponding to its ⟨initiate⟩ in the case of (2) becomes a simple expression shown in Figure 2.11 that only accepts information in many cases.

Then, the exchange units are decided considering speaker's turns and the structure of the unit based on the determined ⟨initiate⟩ part. In many cases, the surface expression of the ⟨follow-up⟩ part becomes "*Hai* (Yes)", "*Wakarimashita* (I understand)" and so on from its definition. So, the ⟨follow-up⟩ is assigned only when these expressions appear

"*Yotei no touroku wo onegai shimasu.*" → [*Request*]
(Registration of the schedule, please.)

"*Mou ichido onegai shimasu.*" → [*Request*]
(Once more please.)

"*Onamae wo onegai shimasu.*" → [*Wh-question*]
(Your name, please.)

Figure 2.12: Examples of different annotation for "*onegai shimasu.*"

in the utterance next to ⟨response⟩. The exchange unit in the remaining portions is determined by heuristic rules.

2.3.4 Inference of Utterance-Unit Tags using Exchange Units

We present the method to decide the utterance-unit tags using the exchange unit described in the previous section.

Decision for Initiate Parts

The tag coming under the ⟨initiate⟩ of the exchange unit is determined on the following procedure. The basic concept is to determine the tags sequentially from the more reliable portion. The procedure is described as follows.

1. Decision of [*Wh-question*] tag
   [*Wh-question*] is annotated for utterances in the ⟨initiate⟩ part containing either an interrogative such as "*Naze* (Why)" or "*Itsu* (When)" or a verb such as "*Oshiete* (tell me)" or "*shiritai* (want to know)".

2. Handling a substitute representation ("*onegai shimasu*")
   The expression "... *wo onegai shimasu.*" in Japanese, which corresponds to "..., please." in English, is usually inferred as [*Request*], but sometimes inferred as [*Wh-question*]. An example is shown in Figure 2.12. We distinguish this by the kind of noun in the preceding "... *wo*" part: if the noun is a common noun, it is considered as [*Request*], otherwise [*Wh-question*]. (When the "... *wo*" part is omitted, they are distinguished by the same judgment on the noun preceding the "onegai shimasu".)

[*Request*] : "... *shite kudasai.*", "... *itadake masuka.*"
[*Confirm*] : "... *ne.*"
[*Yes-No question*] : "... *ka.*"
[*Demand*] : "... *tai n desu kedo.*", "... *tai n desu keredomo.*"
...

Figure 2.13: Examples for end-of-sentence expressions corresponding to each tag

[*Inform*]
A: *Oshiharai kaisuu ha san kai barai de.*
(Would you pay by three times payment?)

[*Positive*]
B: *Hai.*
(Yes.)

Figure 2.14: Example of decision as [*Inform*] by the default rule

There were 32 utterances judged by this criterion in the experimental result, and 29 utterances among them were correctly distinguished.

3. Decision from surface expressions of the end of sentences
   For utterances containing neither the keywords for [*Wh-question*] nor the expression "*onegai shimasu*", utterance-unit tag in the ⟨initiate⟩ part are decided by the surface expressions at the end of the sentence in Figure 2.13.

4. Handling utterances that are not solved by the above
   Remaining utterances that can not determined by the above procedure are decided as [*Inform*]. An example is shown in Figure 2.14.

The ⟨initiate⟩ part in a nested exchange unit also has a function as a response to the outer ⟨initiate⟩ utterance. So, it is regarded as ⟨response/initiate⟩ of the exchange structure shown in Figure 2.3. Therefore, utterance-unit tags for both the ⟨initiate⟩ part and the ⟨response⟩ part are decided respectively following the procedures for each portion.

Decision for Response Parts

The utterance-unit tags coming under the ⟨response⟩ of the exchange unit are often dependent on the kind of utterance-unit tag in the corresponding ⟨initiate⟩. In many cases, the ⟨response⟩ part is directly succeeded by the corresponding ⟨initiate⟩ part. However, there are some exceptions, when the exchange unit is nested as shown in Figure 2.9, for example. In our method, as the exchange unit is decided beforehand, the ⟨initiate⟩ part corresponding to the ⟨response⟩ part can be detected even if the corresponding two utterances are separated.

We classify the ⟨response⟩ tags as follows according to the corresponding ⟨initiate⟩ tags in a task-oriented dialogue.

(a) [*Yes-No question*] → [*Positive*] [*Negative*]

(b) [*Wh-question*] → [*Answer*]

(c) other tags coming under ⟨initiate⟩ → [*Positive*]

These are not always true in general dialogues. But a task-oriented dialogue in this chapter is to solve the task set up beforehand, thus it is expected that participants talk cooperatively. The tendency (b) is often observed because of the assumption. It is also observed that the [*Positive*] tag is assigned for the response part that corresponds to the [*Request*] tag in the corpora.

Therefore, if the ⟨initiate⟩ tag is in the case of either (b) or (c), the ⟨response⟩ tag is decided by the above rules. In the case of (a), the ⟨response⟩ tag becomes either [*Positive*] or [*Negative*].

It is difficult to decide whether the response is either affirmative or negative judged only from the surface expression and the morphological information, when there is no expression showing affirmation or negative directly such as "*Hai.* (Yes.)", "*Sou desu.* (That's right.)" and "... *deha arimasen.* (It is not ...)". For the example of Figure 2.15, utterance #15 denies semantically the content of question #14 although it contains no negative expression.

According to the *maxim of the quantity* in the conversational maxims that H. P. Grice advocated in [23], it is needed that a speaker should tell neither more nor less information. Following this maxim, when the content of a question annotated as [*Yes-No question*] is true, a hearer will be needed only to express affirmation and consequently

[*Yes-No question*]

#14 *Hattori dentetsu tte iu no ha, kuukou no sugu suba ni aru n desyou ka.*
(Is the Hattori Railway immediately near an airport?)

[*Negative*]

#15 *Eki ha kuukou kara sukoshi hanarete imasu.*
(The station is distant a few from the airport.)

Figure 2.15: Example an illocutionary act is [*Negative*] in the response part

the response becomes shorter. On the other hand, when the content of the preceding question is incorrect, the hearer should deny the question and convey correct information alternatively. Consequently, it is expected that the response becomes longer.

In summary, when it cannot be decided whether the response is affirmative or negative from the surface expression and the morphological information, the ⟨response⟩ tag is decided by the following rules.

- If an utterance in the ⟨response⟩ part is longer than 14 characters (in Japanese) → [*Negative* / ⟨initiate⟩]
- Otherwise (shorter than 14 characters) → [*Positive*]

Out of 26 utterances that should be annotated as either [*Positive*] or [*Negative* / ⟨initiate⟩] in corpora, 17 utterances are inferred correctly by the above judgment.

Decision for Follow-up Parts

Utterances coming under the ⟨follow-up⟩ of the exchange unit can be assigned to [*Understand*] by the definition of the utterance-unit tag.

2.4 Experimental Evaluation

We implement the method as the program to infer the utterance-unit tags, and evaluate it experimentally.

2.4.1 Corpora and Conditions for Experiment

We select four tasks of group scheduling, route direction, and telephone shopping and travel information included in CD-ROM of "Simulated Spoken Dialogue Corpus" in the

Table 2.1: Amount of training data and accuracy of the inference (closed test)

		# dialogue	# utterance	accuracy (%)
total		15	611	86
breakdown	group scheduling	10	340	89
	route direction	2	69	72
	telephone shopping	3	202	86

Table 2.2: Amount of test data and accuracy of the inference (open test)

	# dialogue	# utterance	accuracy (%)
group scheduling	4	169	79
route direction	1	62	70
telephone shopping	1	75	84
travel information	1	68	53
total	7	374	73

Grant-in-Aid for Scientific Research on Priority Areas [24]. We pre-process these corpora for the annotation by dividing and combining into the annotation unit, and annotate utterance-unit tags by hand as correct answers. Then, we split the corpora into training data and test data, and described the rules shown in section 2.3 from all the training data. The inference is executed for both the training data and the test data. The accuracy is calculated by comparing the output of the program with the correct tag annotated by hand in advance.

2.4.2 Experimental Results

Table 2.1 shows the amount of training data used to describe rules and the accuracy of the inference for them (closed test). In the closed test, accuracy of 86% is achieved for 611 utterances.

The result of the open test is shown in Table 2.2. Accuracy in the open test is 73% on the average for 374 utterances.

2.4.3 Discussions

The rules extracted from training data are not dependent on the knowledge peculiar to the task, but the inference program utilizes the characteristics that are dependent on the speaker's style such as expressions of the end of sentences. Thus, the accuracy is higher in the tasks having more training data in the experimental result. Especially in

B: "*Dobashi to iu hashi wo higashi gawa ni watatte, sugu migi ni magatte kudasai.*"
(Cross a bridge called Dobashi to the east side, and turn right immediately.)

A: "*Hai.*" (All right.)

Figure 2.16: Example of incorrect inference

the task-oriented dialogue, it is expected that the expressions of the questions to achieve the goal becomes a fixed form. If sufficient training data are available, accuracy will be close to the upper limit of this method. The low accuracy in the open test of the travel information task suggests insufficiency of the amount of training data. It is necessary to have expressions of various speakers, especially for the end of sentences sufficiently.

In the closed test, the accuracy of the route direction task is lower than other two tasks. This is because there are some portions that need the special knowledge in the route direction task. The example is shown by sample sentences in Figure 2.16. The utterance of speaker 'B' in this example shows the interlocutor the route to the destination, and then this utterance should be annotated as [*Inform*]. However, the output of the program becomes [*Request*] because of the end-of-sentence expression "*shite kudasai*". In such a case, a human uses not only the information extracted from utterances but also the background knowledge that the task is route direction. Except for these cases, the proposed method works effectively in general.

2.5 Conclusions

In this chapter, we present a method to infer the utterance-unit tag, which is one of the discourse tags for spoken dialogue corpora. The input to the inference program is limited only to surface text and the morphological analysis result. The inference is performed by using the characteristic expressions and the exchange structure without using the knowledge dependent on the task. The proposed method can be applied to annotate the utterance-unit tags in various tasks if more training is performed and the variety of expressions are further accumulated.

Accuracy of the tagging is 86% for the closed test, and 73% for the open test. In the task where the feature of speaker's expression can be extracted with sufficient training data (group scheduling and telephone shopping), accuracy of nearly 80% can be obtained

even in the open test. This result shows that the program based on the method can play both roles to support the annotation of the utterance-unit tag.

Chapter 3

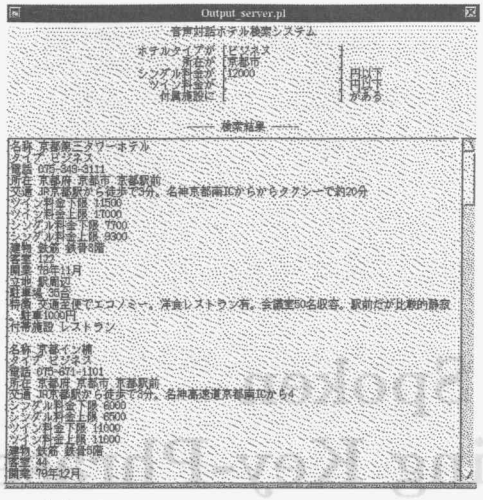
Domain-Independent Spoken Dialogue Platform using Key-Phrase Spotting based on Combined Language Model

3.1 Introduction

With improvement of speech recognition technology, many kinds of spoken dialogue systems have been developed. Information query is regarded as one of the most promising tasks, because the majority of operations in this task are to select from huge entries, in which speech interface is advantageous. But spoken dialogue systems are not ubiquitous yet. One of the causes is lack of portability [25][26] . It is necessary to set up appropriate linguistic constraints and semantic interpretation rules, but it requires a great deal of labor with expertise. When we use elaborate corpus-based statistic models, it is necessary to collect large amount of a corpus that matches the task and domain. Thus, rapid prototyping technique is important in data collection as well as system development.

Therefore, we develop a domain-independent platform for information query tasks. Domain-dependent information is extracted from the domain database. General rules for information query are retained as domain-independent information. By limiting the task to typical information query, a lexicon and grammar rules for speech recognition are extracted from the domain database based on a simple task description.

The other serious problem in spoken dialogue system is recognition errors caused by out-of-vocabulary and out-of-grammar utterances. A novice user often makes utterances beyond system's capacity, since they do not know the acceptable vocabulary and gram-



(a) A real system in Japanese (b) Upper portion translated in English

Figure 3.1: Outlook of GUI for information query

mar. In order to accomplish the system's task successfully, users should be guided into acceptable expressions. We adopt a GUI that shows patterns of acceptable utterances and current query status explicitly, so that user utterance can be guided into the system's capacity. Moreover, our key-phrase spotter that utilizes the generated grammar and 2-gram model trained with similar domain corpora can cope with various utterances and extract key-phrases flexibly.

3.2 Approach to Domain-Independency

A typical spoken dialogue system consists of speech recognition, semantic interpretation and dialogue management modules. In order to realize full domain-independency, these three steps must be domain-independent. But unlimited-vocabulary spontaneous speech recognition and universal semantic interpretation for any domains are very difficult problems[27].

In our platform, we assume that the system is to perform information query using multi-modal interfaces. Information query can involve a lot of domains, but provides constraint to a semantic analyzer so that key-phrase-based understanding is feasible. Use of multi-modal interfaces with a display eases the problems of speech recognition and dialogue management.

3.2.1 Model of Information Query

For generality and simplicity, we regard information query as filling a query form that consists of a set of search keys. Thus, user utterances are modeled as setting and retracting search keys. Domain of the query is not limited, but includes trains, flights and hotels.

The platform semi-automatically generates a lexicon and grammar rules that cover possible expressions of search keys. They are derived from the domain database, i.e. database of trains or hotels. Search keys are generally made of a set of search items and their values, for example, "location is Tokyo", which usually correspond to database fields and entries, respectively. Thus, a lexicon is automatically derived from the database fields and entries. A baseline grammar within key-phrases is also set up to accept typical expressions used in the query.

In this framework, the semantic analysis is achieved as filling query slots with keywords and translating key-phrases into them. It is independent of domains. We also prepare several universal patterns for expressions to retract or clear search keys.

3.2.2 Use of GUI

Use of a GUI (Graphical User Interface) constrains user utterances to be recognized and complements a dialogue manager.

Among the major problems in speech understanding is out-of-vocabulary or out-of-grammar expressions. On the other hand, we have observed that users often hesitate to speak to machines simply because they do not know which forms of expressions are acceptable. Our platform displays key-phrase patterns as a visual form of the query, which guides how to speak to users and reduces the variation of input utterances.

Moreover, the system promptly displays recognition results in the slots of the visual query form as well as the query results. The feature lets the users know recognition errors and eliminates the necessity of confirmation through spoken dialogue. Instead, the users simply make "undo" commands in case of errors. This feature will avoid possible crashes in dialogue. It also enables users specify preferences incrementally by reviewing the current (number of) matched entries. It is useful for those who do not have in mind a definite preference beforehand.

An outlook of the GUI is shown in Figure 3.1.

3.2.3 Portability of Language Model for Speech Recognizer

N-gram model is a powerful language model of the speech recognizer if sufficient training corpus of particular domain is available, but it is difficult to collect sufficient amount of corpus for every specific domain. On the other hand, a manual grammar is often used as language model of spoken dialogue systems. It does not need training corpus and can introduce domain-specific knowledge easily. But it is nearly impossible to describe all expression patterns with grammar rules because user utterances have enormous variations especially in filler portions[1]. Speech recognition using a described grammar only is often too rigid.

Kawahara et al. proposed a method based on key-phrase spotting to recognize spontaneous speech flexibly[28]. The grammar rules solely for the key-phrase portions are definite and simple, so can be written with less labor accordingly. It realizes robustness against ill-formed utterances. However, the constraint of key-phrase grammars without statistics is so loose that false alarms consisting of short words appear frequently.

Considering such issues, we present a novel phrase-spotting method based on combined language model, which consists of both grammar rules for domain-dependent key-phrases and 2-gram constraint derived from similar domain corpora for domain-independent fillers. The model puts proper constraint over the whole sentence by applying N-gram to filler portions, and consequently can improve the recognition accuracy. As word N-gram model is trained with similar domain corpora that do not necessarily match the query domain, our method realizes portability for various domains.

3.3 Spoken Dialogue Platform

The platform consists of two parts: task specification tool and domain-independent spoken dialogue engine as shown in Figure 3.2. The task specification tool generates a lexicon and grammar rules based on a task specification by an application developer. The domain-independent spoken dialogue engine works as an interface with a user according to the task description files. The task specification tool and the domain-independent spoken dialogue engine are explained in section 3.3.1 and section 3.3.2, respectively.

[1]We regard the portions other than key-phrases as filler, such as "I would like".

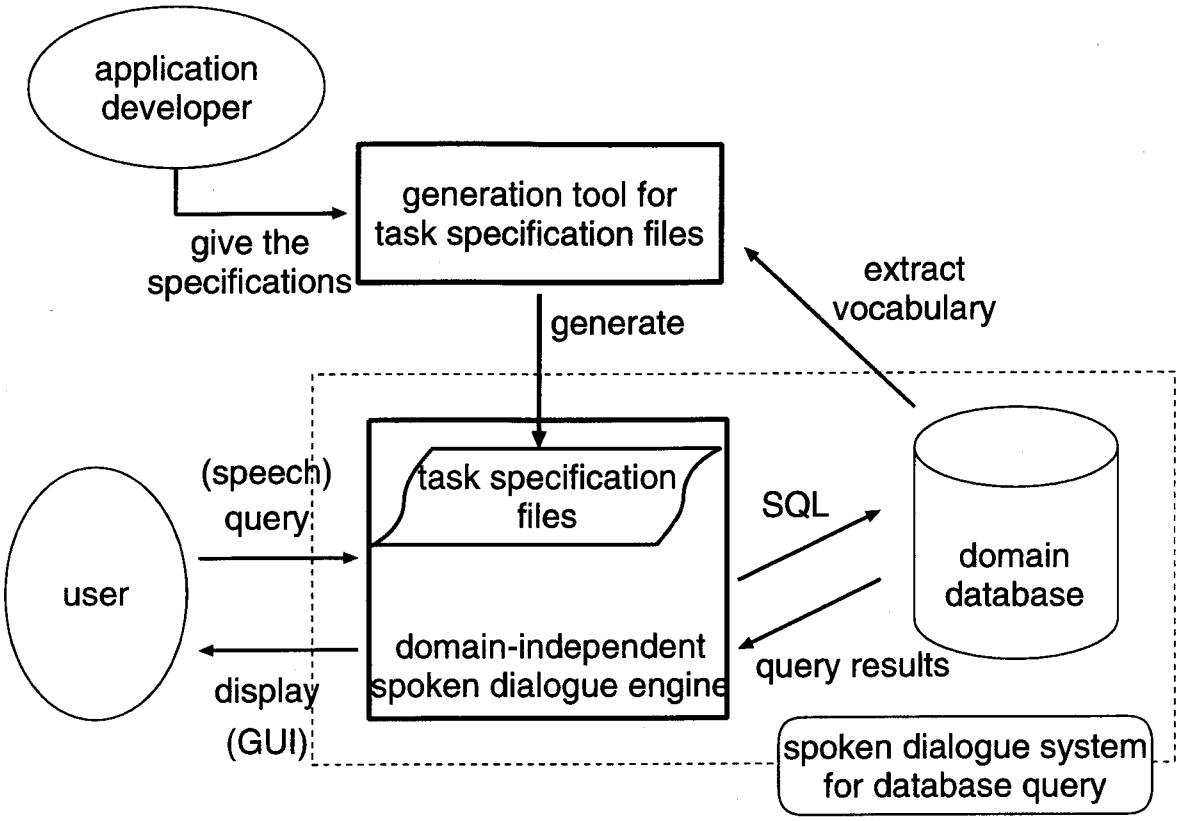


Figure 3.2: Overview of domain-independent platform of spoken dialogue interface

3.3.1 Task Specification Tool

The task specification tool creates task description files by generating grammar rules and extracting the vocabulary from the target domain database based on the task specification by an application developer. The flow of this process is shown in Figure 3.3. The task description files contain grammar rules for semantic interpretation, a homonym list and a set of query items as well as a lexicon and grammar rules of key-phrase parts for a speech recognizer.

The GUI of the task specification tool is shown in Figure 3.4. An application developer defines patterns of key-phrases with the GUI by specifying names of query items, aliases of the items, post-positional particles following keywords, aliases of the particles, the unit and title of keywords and so on. Furthermore, an extraction method of keywords, a phrase grammar type (whether to accept the omission of a name of query items or not), and a query type (how to generate an SQL when a keyword is input) are selected from items listed in the GUI. The extraction method of keywords is selected from either

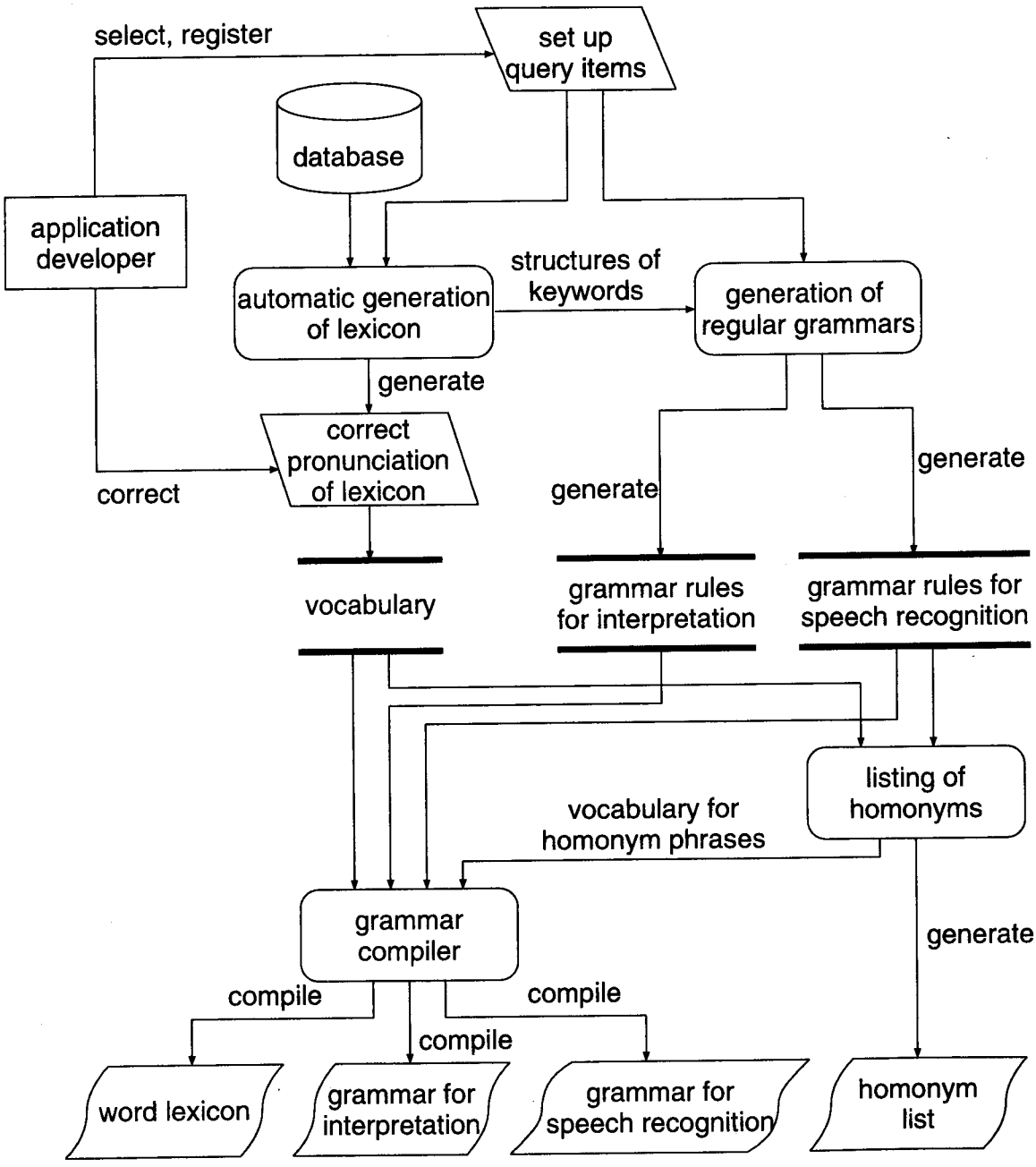


Figure 3.3: Overview of generation of task description files

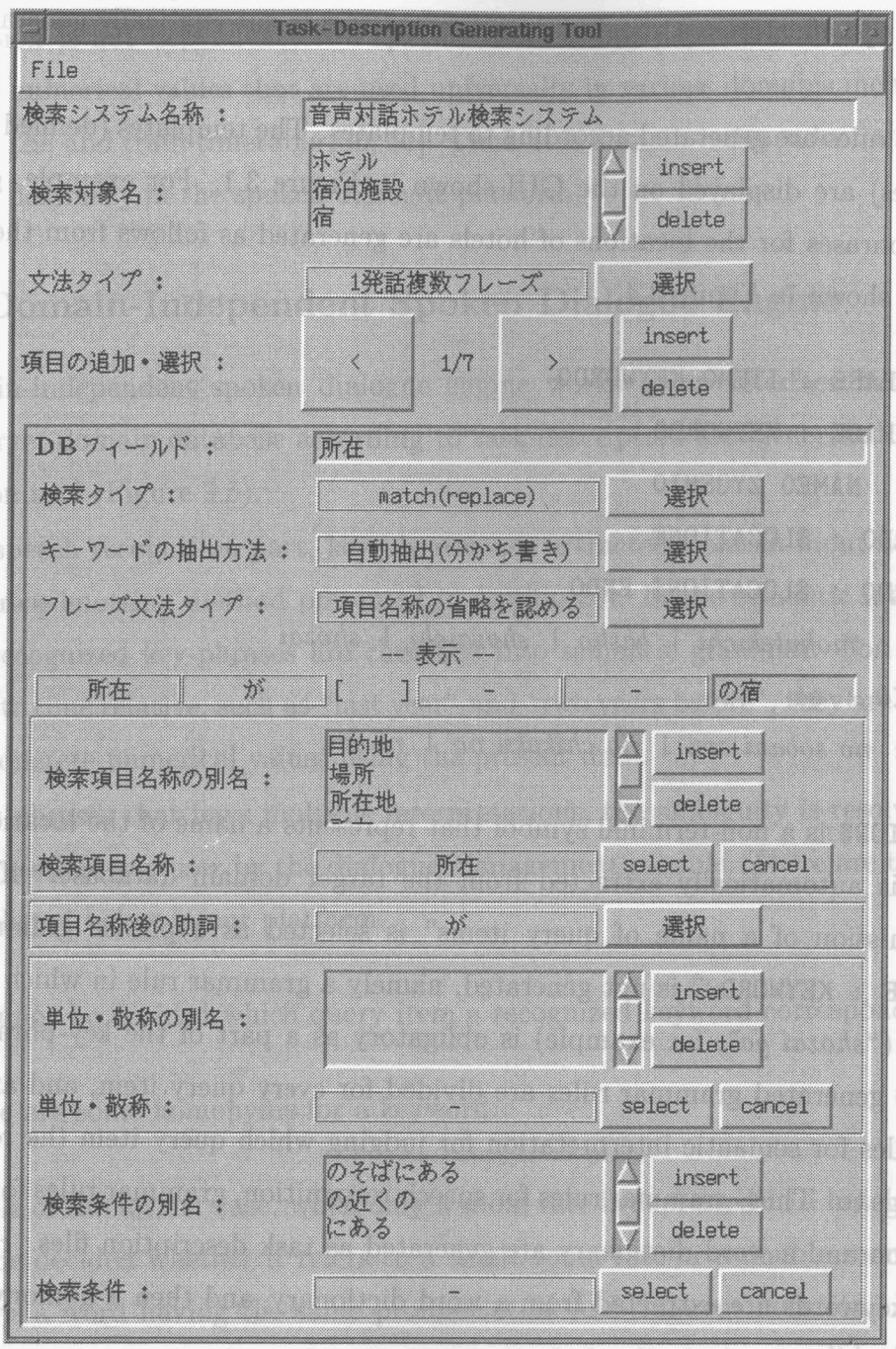


Figure 3.4: Outlook of GUI for task description

automatic extraction from a database (either extracting the database entries delimitated with spaces just as they are, or extracting after segmentation with the morphological analysis) or generation from the concept such as a numerical value, date and year. Based on these task specifications, a vocabulary for every item is automatically generated from a database.

Grammar rules are generated according to templates. The templates (defined patterns of key-phrases) are displayed on the GUI shown in Figure 3.1. For example, grammar rules of key-phrases for the locations of hotels are generated as follows from the content of the screen shown in Figure 3.4.

KEY_PHRASE : ITEM0 KEYWORD0
KEY_PHRASE : KEYWORD0
ITEM0 : NAME0 ZYOSHI0
KEYWORD0 : $LOCATION$
KEYWORD0 : $LOCATION$ END0
NAME0 : *mokutekichi* | *basho* | *shozaichi* | *shozai*
ZYOSHI0 : *ga*
END0 : *no sobani aru* | *no chikaku no* | *ni aru*

The $LOCATION$ is a non-terminal symbol that represents a name of the location, whose vocabulary is automatically extracted from the target domain database. If "not accept the omission of a name of query items" is selected as a phrase grammar type, "KEY_PHRASE : KEYWORD0" is not generated, namely a grammar rule in which the name of the item ("*shozai ga*", for example) is obligatory as a part of the key-phrase is generated. The generated grammar rules are divided for every query item, and are used as grammar rules for semantic interpretation for judging which query item the recognition result belongs to. Thus, grammar rules for speech recognition, grammar rules for semantic interpretation and a word dictionary are generated as task description files. Homonyms in a set of keywords are extracted from a word dictionary, and then a homonym-list file is generated.

The generated dictionary is expanded for the name of a place and numerical expression when the vocabulary is extracted from the database. For example, even if the entry of "8000 yen" is not included in the entries of database by chance, the tool generates the lexicon to cover possible values. Specifically, it cope with the numerical fields, such as

"price", "distance" and fields of dates such as "year", "date", "time" and "day of the week". The application developer specifies like "ranging by 1000 yen from 1000 yen to 20000 yen". Then, the platform will generate the lexicon according to this specification. The extension is not dependent on a specific domain but dependent on the concept of the date or numerical values that are used universally in various domains including flight reservation [29] and train timetable [30, 31]. Thus, it is domain-independent and expected to improve flexibility in the spoken dialogue platform.

3.3.2 Domain-Independent Spoken Dialogue Engine

The domain-independent spoken dialogue engine works as an interface between users and the target domain database according to task description files generated by the task specification tool (Figure 3.5).

In the speech recognition part, key-phrases are extracted from an input utterance by the key-phrase spotting method presented in section 3.4. In the semantic interpretation part, the recognized key-phrases are classified into semantic grammar. For expressions that indicate time relative, such as "last year" and "two years before", they are interpreted here into concrete numerical values using the present date.

For key-phrases that have multiple interpretations, the ambiguity is resolved by generating a question to a user by the dialogue management module. There are two kinds of ambiguity listed below in our platform.

(1) It cannot be decided which query item a recognized keyword corresponds to.

(2) There are some homonyms for a keyword.

For example of the former case, when only a room rate is expressed in a hotel query task, it cannot be decided whether it refers to a single room rate or a double room rate. The homonym is a word having the same pronunciation and different notation. In the both cases, the system resolves the ambiguity interactively by displaying candidates to a user.

The query conditions are updated according to the obtained semantic interpretation results, and then the query is executed. The query condition and the query results are displayed to a user immediately and interactively with the GUI. As the query results, the number of the retrieved items and all their contents are presented to a user.

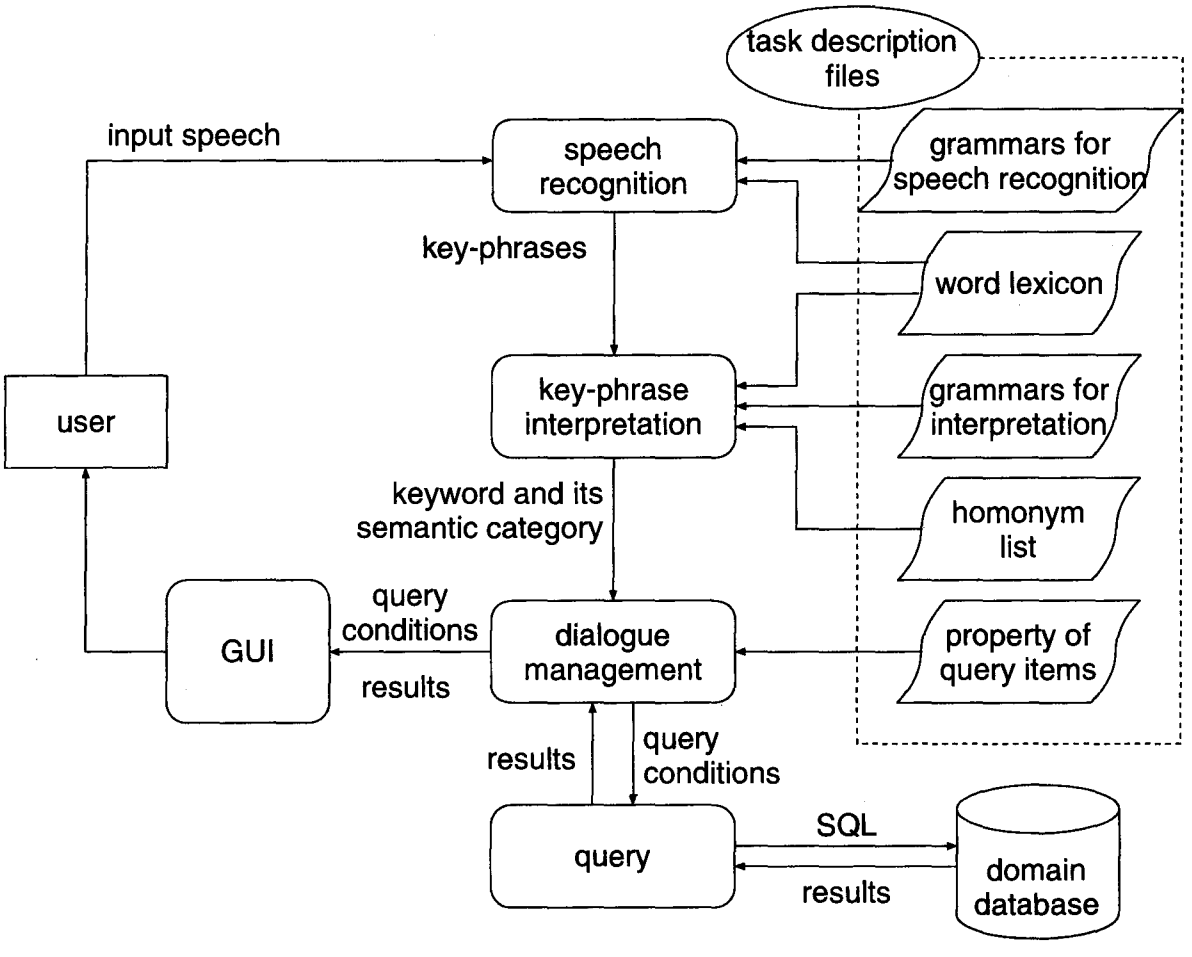


Figure 3.5: Overview of domain-independent spoken dialogue engine for database query

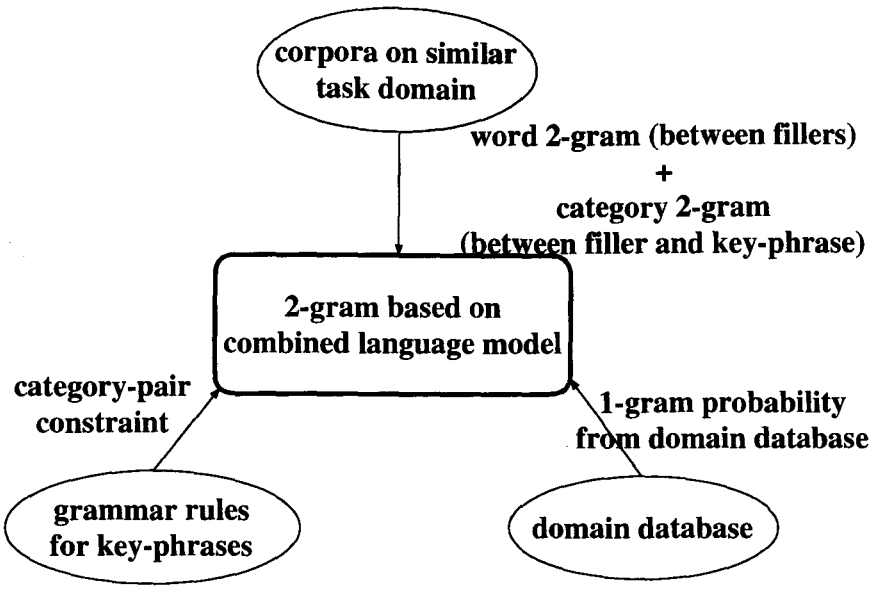


Figure 3.6: Concept of combined language model

3.4 Key-Phrase Spotting based on Combined Language Model

Next, we address the language model for speech recognition.

3.4.1 Combination of Grammar Rules and Statistical Model

We assume that the training corpus is not perfectly matched to the domain is not available, but those similar to the system's task are available. Based on this assumption, we construct the linguistic constraint from three information sources (Figure 3.6).

As a constraint between fillers that does not affect key-phrase portions directly, we apply 2-gram probabilities estimated with similar domain corpora, since filler portions are regarded as domain-independent. In key-phrases, word transitions are defined based on the category-pair constraint that is automatically derived from the generated phrase grammar.

As 2-gram model between a filler and a key-phrase, we introduce a class 2-gram that consists of nouns, which make up key-phrases. For words in key-phrases that have relatively domain-independent concepts (price, date, name of place), a specific class is prepared. Moreover, when this class 2-gram probabilities is transformed into word 2-gram probability, 1-gram probability is provided based on the distribution of domain database

entries.

Specifically, the constructed language model is formulated as below. Here, $p_{co}()$, $p_{db}()$ and $p_{gr}()$ denote the probability derived by similar domain corpora, a domain database and a phrase grammar, respectively. The probability assigned to the phrase grammar ($p_{gr}(c_2|c_1)$) is 1 if the concatenation c_1c_2 is defined in grammar rules, otherwise 0.

- for filler portions
  $$p(w_2|w_1) = p_{co}(w_2|w_1)$$
- between key-phrase and filler
  $$p(w_2|w_1) = p_{db}(w_2|c_2) \cdot p_{co}(c_2|w_1)$$
- inside key-phrase
  $$p(w_2|w_1) = p_{db}(w_2|c_2) \cdot p_{gr}(c_2|c_1)$$
  $$= \begin{cases} p_{db}(w_2|c_2) & (\text{if } c_1c_2 \text{ is defined}) \\ 0 & (\text{otherwise}) \end{cases}$$

For example, a probability between a key-phrase "Kyoto" and a filler "in" p(Kyoto | in) is calculated by multiplying p_{co}(PLACE | in) derived from similar corpora by p_{db}(Kyoto | PLACE) derived from the domain database.

3.4.2 Key-Phrase Spotting based on Combined Model

We adopt a progressive search strategy focused on key-phrases portions. It applies more strict linguistic constraint incrementally on key-phrases in the three steps (Figure 3.7).

1st pass word 2-gram model and category-pair grammar

2nd pass phrase grammar (inside key-phrase) and word 2-gram model

3rd pass inter-phrase grammar (when connecting key-phrases)

In the 1st and 2nd passes, key-phrases are spotted based on the combined language model derived from the word 2-gram and grammar rules. In the 3rd pass, we connect spotted phrases and recognize as a sentence. In this step, phrase candidates are connected according to their scores and semantic constraint, which is defined as inter-phrase grammar rules. Because spotting methods do not assume parsing the whole sentence, we put a constant penalty value when there is a skipped portion between key-phrases.

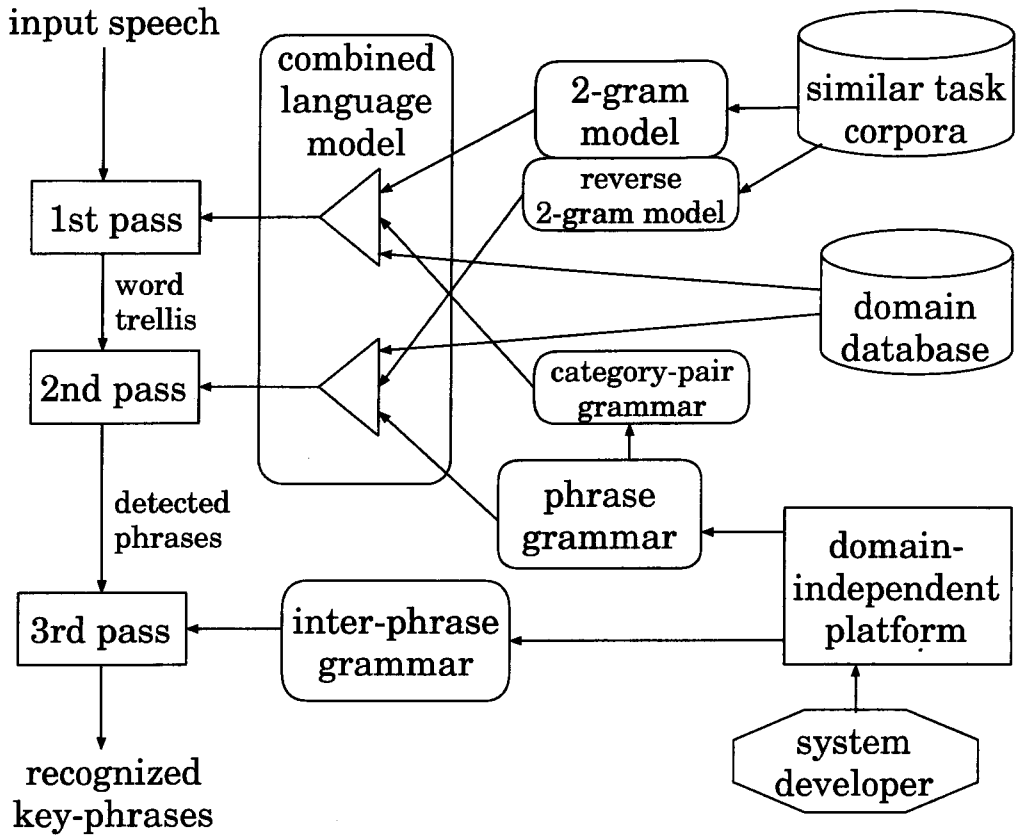


Figure 3.7: Overview of key-phrase spotting method

3.5 Experimental Evaluation

We have applied the platform to hotel database and literature database, and constructed a hotel query system and a literature query system, respectively. The hotel database contains 2040 entries and has seven items such as location, upper limit of room rate, facilities and so on. Users can specify and retract slot values corresponding to the items. It is possible to specify and retract multiple items in a single utterance. The literature database contains 1913 entries and has five items users can specify, such as title, author(s), journal name and published year. Experimental evaluation is made on the hotel query system.

3.5.1 Initial Performance of Generated System for Hotel Domain

Performance of portable platforms can be measured by the coverage of the system over user utterances. We evaluate robustness of our platform by counting how user utterances

Table 3.1: Performance of the generated grammar compared with hand-crafted grammar

	generated grammar	hand-crafted grammar
user utterances within system capacity	93%	55%
recognition accuracy	69%	31%

are accepted by the prototype system. We make two experiments using the same speech recognizer Julian [32]. The vocabulary of the prototype system contains 982 words, which have been extracted from the hotel database automatically. The grammar used in this experiment is a finite state automaton that is repetition of key-phrases.

In the first experiment, we use two hotel query systems: one is the prototype system generated by our platform and the other uses a carefully hand-crafted grammar for the speech recognizer. Subjects are students in our department. In the prototype system, acceptable patterns of utterances are displayed explicitly. The ratio of acceptable utterances is shown in Table 3.1. Better performance is achieved by the generated grammar. This result demonstrates that both robustness and portability can be achieved by using a simple generated grammar of key-phrases and guiding user utterances rather than writing a complex sentence-level grammar manually.

In the second experiment, we evaluate the effect of guidance by GUI for novice users using the generated grammar. Subjects are 28 novice users (19 males and 9 females) who have not used a spoken dialogue system before. We set up dialogue condition 1 and 2 described below. For about five minutes, each user tries the system without given scenario.

condition 1 Users are given a manual showing search keys (location, hotel type, room rate, facilities and so on) beforehand.

condition 2 Acceptable utterance patterns and the recognition results are displayed to the user through the GUI (Figure 3.1).

There are 518 and 429 utterances in condition 1 and 2, respectively. Table 3.2 lists their comparison. The ratio of user utterances within the system's capacity is significantly larger in condition 2. This result confirms that the proposed GUI works very effectively

Table 3.2: Classification of user utterances by the effect of GUI

	condition 1	condition 2
utterances within system's capacity	44.6%	76.4%
# out-of-vocabulary	21.8%	11.4%
# out-of-grammar	2.1%	1.2%
# out-of-task	31.5%	11.0%

Table 3.3: Performance of our method compared with two conventional methods

	vocabulary size	in-grammar	nearly-in-grammar	out-of--grammar	total
# correct keywords		561	116	120	797
grammar rules for the whole sentence	942	14.8%	51.4%	175.2%	45.8%
phrase spotting (without 2-gram)	1290	16.8%	34.2%	154.2%	37.9%
phrase spotting (with 2-gram)	6124	**10.8%**	**24.7%**	**140.9%**	**30.3%**

FA: ratio of incorrectly recognized keywords
SErr: ratio of keywords that are not recognized

as a guidance of user utterances.

In summary, the portability is maintained by generating a simple phrase grammar without spoiling the robustness, which is enhanced by the use of GUI. The system is proved to suffice the prototype for data collection.

3.5.2 Improvement by Combined Language Model

Next, we implement the combined language model and compare it with the conventional methods. We construct the combined language model using 2-gram model trained with the ATR [33, 34] and RWC [35] corpora. The corpora consist of dialogues at travel agents, hotel receptions and car dealers, and the tasks are not identical to the system's one. The text size is about 208 thousands and the vocabulary size is 5432 in total.

Subjects are 24 novice users (19 males and 5 females). As a test set, we use 665

utterances collected using the prototype hotel query system. A gender-dependent triphone model is used as the acoustic model. As an evaluation measure, we use the sum of the ratio of incorrectly recognizing keywords (False Acceptance: FA) and the ratio of slots that are not filled with correct values (Slot Error: SErr). Namely, FA and SErr are defined as the complements of the precision rate and the recall rate, respectively.

$$FA = \frac{\#\text{ of incorrectly recognized keywords}}{\#\text{ of recognized keywords}}$$

$$SErr = \frac{\#\text{ of incorrectly recognized keywords}}{\#\text{ of all correct keywords}}$$

The test set samples are classified into three types: in-grammar, nearly-in-grammar and out-of-grammar. Table 3.3 lists the "FA+SErr"[2] of our proposed method and two conventional methods. One uses grammar rules for the whole sentence and the other adopts key-phrase spotting without the 2-gram model.

For in-grammar samples, the proposed method using the combined language model gets the best performance. Use of the 2-gram model suppresses the false alarms. Since this 2-gram model is trained with similar domain corpora, the phrase spotting even outperforms the full sentence grammar while maintaining the portability. For both nearly-in-grammar and out-of-grammar samples, the best performance was also achieved by the combined language model. Even for ill-formed utterances, the proposed method realizes robust understanding. In total, the semantic accuracy is improved by 15.5%.

The key-phrase spotting approach is superior to grammar-based approach, especially for ill-formed utterances. And the superiority is enhanced by introducing 2-gram model that does not necessarily match the query domain. Thus, this improvement does not spoil the system's portability.

3.6 Conclusions

We have presented a portable spoken dialogue platform for information query and its experimental evaluation. The platform can be applied to various domains because it generates domain-dependent lexicon and grammar rules extracted from the domain database automatically. This portability of language model is realized by adopting simple key-phrase spotting strategy. It is also enhanced by incorporating statistics derived from

[2]Substitution errors are counted both as FA and SErr.

similar domain corpora and the domain database. Moreover, we make use of GUI that displays typical acceptable patterns, which guides users within the system's lexicon and grammar. Overall strategy is demonstrated to work as a reasonable prototype system and realize even robust understanding for ill-formed utterances. The proposed framework does not need collecting domain-specific corpus or writing grammar rules. Thus, it is a domain-independent platform.

Chapter 4

Generating Confirmation and Guidance using Two-Level Confidence Measures of Speech Recognition Results

4.1 Introduction

In a spoken dialogue system, it frequently occurs that the system incorrectly recognizes user utterances and the user makes expressions the system does not expect. These problems are essentially inevitable in handling the natural language by computers, even if vocabulary and grammar of the system are tuned. This lack of robustness is one of the reasons why spoken dialogue systems have not been widely deployed.

In order to realize a robust spoken dialogue system, it is inevitable to handle speech recognition errors. To suppress recognition errors, system-initiative dialogue is effective. But it can be adopted only in a simple task. For instance, the form-filling task can be realized by a simple strategy where the system asks a user the slot values in a fixed order. In such a system-initiated interaction, the recognizer easily narrows down the vocabulary of the next user's utterance, thus the recognition gets easier.

On the other hand, in more complicated tasks such as information retrieval, the vocabulary of the next utterance cannot be limited on all occasions, because the user should be able to input the values in various orders based on his preference. Therefore, without imposing a rigid template upon the user, the system must behave appropriately even when speech recognizer output contains some errors.

Obviously, making confirmation is effective to avoid misunderstandings caused by

speech recognition errors. However, when confirmation is made for every utterance, the dialogue will become too redundant and consequently troublesome for users. Previous works have shown that confirmation strategy should be decided according to the frequency of speech recognition errors, using mathematical formula [5] and using computer-to-computer simulation [7]. These works assume fixed performance (averaged speech recognition accuracy) in whole dialogue with any speakers. For flexible dialogue management, however the confirmation strategy must be dynamically changed based on the individual utterances. For instance, we human make confirmation only when we are not confident. Similarly, confidence measures (CM) of speech recognition output should be modeled as a criterion to control dialogue management.

CMs have been calculated in previous works using transcripts and various knowledge sources [36, 37]. For more flexible interaction, it is desirable that CMs are defined on each word rather than whole sentence, because the system can handle only unreliable portions of an utterance instead of accepting/rejecting whole sentence.

In this chapter, we propose two concept-level CMs on content-word level and on semantic-attribute level for every content word. The system can make efficient confirmation and effective guidance according to the CMs. Even when successful interpretation is not obtained on content-word level, the system generates system-initiative guidance based on the semantic-attribute level, which will lead the next user's utterance to successful interpretation.

4.2 Definition of Confidence Measures (CM)

Confidence Measures (CMs) have been studied for utterance verification that verifies speech recognition result as a post-processing [28]. Since automatic speech recognition is a process finding a sentence hypothesis with the maximum likelihood for an input speech, some measures are needed in order to distinguish a correct recognition result from incorrect ones. In this section, we describe definition of two level CMs, which are on content-words and on semantic-attributes, using 10-best output of the speech recognizer and parsing with phrase-level grammars.

4.2.1 Definition of CM for Content Word

In the speech recognition process, both acoustic probability and linguistic probability of words are multiplied (summed up in log-scale) over a sentence, and the sequence having the best likelihood is obtained by a search algorithm. The score of sentence derived from the speech recognizer is a log-scaled likelihood of a hypothesis sequence. We use a grammar-based speech recognizer Julian [32], which was developed in our laboratory. It correctly obtains the N-best candidates and their scores by using A* search algorithm.

Using the scores of these N-best candidates, we calculate content-word CM as below. The score of each sentence output by the recognizer is a log-scaled likelihood. The content words are extracted by parsing with phrase-level grammars that are used in speech recognition process. We set $N = 10$ after we examined various values of N as the number of generated candidates. Even if we set N larger than 10, the scores of i-th hypotheses ($i > 10$) are too small to affect resulting CMs.

First, each i-th score is multiplied by a factor α ($\alpha < 1$). This factor smoothes the difference of N-best scores to get adequately distributed CMs. Because the distribution of the absolute values is different among kinds of statistical acoustic model (monophone, triphone and so on), different values must be used. The value of α is examined in the preliminary experiment. We set $\alpha = 0.05$ when using a triphone model as acoustic model. Next, they are transformed from a log-scaled value ($\alpha \cdot scaled_i$) to probability dimension by taking its exponential, and calculate a posteriori probability for each i-th candidate [38].

$$p_i = \frac{e^{\alpha \cdot scaled_i}}{\sum_{j=1}^{n} e^{\alpha \cdot scaled_j}}$$

This p_i represents a posteriori probability of the i-th sentence hypothesis.

Then, we compute a posteriori probability for a word. If the i-th sentence contains a word w, let $\delta_{w,i} = 1$, and 0 otherwise. A posteriori probability that a word w is contained (p_w) is derived by summation of a posteriori probabilities of sentences that contain the word.

$$p_w = \sum_{i=1}^{n} p_i \cdot \delta_{w,i}$$

We define this p_w as the content-word CM (CM_w). This CM_w is calculated for every content word. Intuitively, words that appear many times in N-best hypotheses get high CM, and frequently substituted ones in N-best hypotheses are judged as unreliable.

utterance: "*oosakafu no singururyoukin ga 19000 en no yado*"
("Tell me hotels in Osaka-pref. less than 19000 yen for a single room.")

i	Recognition candidates (<g>: filler model)	$score_i$	p_i
1	*oosakafu no singururyoukin ga 19000 en ika no* <g> Osaka-pref.(location) / less than 19000 yen for a single room	-16490	.15
2	*oosakafu no singururyoukin ga 19000 en ika no yado* Osaka-pref.(location) / less than 19000 yen for a single room	-16493	.13
3	*oosakafu no singururyoukin ga 12000 en ika no* <g> Osaka-pref.(location) / less than 12000 yen for a single room	-16495	.12
4	*oosakafu no singururyoukin ga 18000 en ika no* <g> Osaka-pref.(location) / less than 18000 yen for a single room	-16496	.11
5	*oosakafu no singururyoukin no 12000 en ika no yado* Osaka-pref.(location) / less than 12000 yen for a single room	-16498	.10
6	*oosakafu no singururyoukin ga 14000 en ika no* <g> Osaka-pref.(location) / less than 14000 yen for a single room	-16498	.10
7	*oosakafu no singururyoukin ga 18000 en ika no yado* Osaka-pref.(location) / less than 18000 yen for a single room	-16500	.09
8	*oosakafu no singururyoukin no 16000 en ika no* <g> Osaka-pref.(location) / less than 16000 yen for a single room	-16501	.09
9	*oosakafu no singururyoukin no 14000 en ika no yado* Osaka-pref.(location) / less than 14000 yen for a single room	-16502	.08
10	*oosakashi no singururyoukin no 19000 en ika no* <g> Osaka-city.(location) / less than 19000 yen for a single room	-16518	.04

CM_w	(word)@(attribute)
0.96	Osaka-pref.@location
0.31	19000yen@single:max
0.22	12000yen@single:max
0.20	18000yen@single:max
0.18	14000yen@single:max
0.09	16000yen@single:max
0.04	Osaka-city.@location

CM_c	semantic attribute
1.00	single:max
0.50	location

Figure 4.1: Example of calculating CM

In Figure 4.1, we show an example in CM_w calculation with recognizer outputs (i-th recognized candidates and their a posteriori probabilities) for an utterance "*oosakafu no singururyoukin ga 19000 en ika no yado* (Tell me hotels in Osaka-pref. less than 19000 yen for a single room.)". It is observed that a correct content word 'Osaka-pref. as location' gets a high CM value ($CM_w = 1$). The others, which are incorrectly recognized, get low CM values, and shall be rejected.

4.2.2 CM for Semantic Attribute

A concept category is defined as a semantic attribute assigned to content words, and it is identified by parsing with phrase-level grammars that are used in speech recognition process and represented with Finite State Automata (FSA). Since these FSAs are classified into concept categories beforehand, we can automatically derive the concept categories of words by parsing with these grammars. In the hotel query task, there are seven concept categories such as 'location' and 'facility'.

For this concept category, we also define semantic-attribute CM (CM_c). First, we calculate a posteriori probabilities of N-best sentences in the same way of computing content-word CM. Here, we introduce $\beta_{c,i}$ representing likelihood that a phrase $W_{c,i}$ belongs to a category c in the i-th sentence.

First, a phrase containing words that appear only in a specific category has a higher likelihood belonging to the category c than that consists of general words such as numerical expressions. We introduce this preference by using the idf (inverse document frequency) values. An idf value for a word w_j is defined as below.

$$idf(w_j) = \log \frac{M}{df(w_j)}$$

M denotes the total number of semantic categories, and the $df(w_j)$ denotes the number of categories where w_j appears. Namely, an idf value of a word appearing only in a specific category becomes large.

Next, a phrase specifying the name of its item like "*Shozai ga Kyoto-fu no* (located in Kyoto-pref)" has also higher likelihood belonging to the category (location, in this example) than a short one that consists of only a content word and its post-positional particle like "*Kyoto-fu no* (in Kyoto-pref)". So, $\beta_{c,i}$ should be larger if a phrase $W_{c,i}$ is longer, namely the number of contained words is larger.

Based on the above discussion, we define $\beta_{c,i}$ that represents a likelihood that phrase $W_{c,i}$ belongs to category c, by calculating the summation of idf values of words contained in the phrase $W_{c,i}$, and normalizing by γ_c, which is set for each category.

$$\beta_{c,i} = \frac{1}{\gamma_c} \sum_{w_j \in W_{c,i}} idf(w_j)$$

γ_c is the maximum value of $\beta_{c,i}$ calculated beforehand for possible phrases that can appear in category c. We define the semantic-attribute CM (CM_c) by the summation of the product of $\beta_{c,i}$ and a posteriori probability of the i-th sentence as below.

$$CM_c = \sum_{i=1}^{N} \beta_{c,i} \cdot p_i$$

4.3 Dialogue Management using Confidence Measures

There are a lot of systems that adopt a mixed-initiative strategy [31, 39, 40]. It does not impose rigid system-initiated templates and a user can specify what he/she has in mind directly, thus the dialogue becomes more natural. In conventional systems, the system-initiated utterances are considered only when semantic ambiguity occurs. But in order to realize robust interaction, the system should make confirmation to remove recognition errors and generate guidance to lead next user's utterance to successful interpretation. In this section, we describe how to generate the system-initiated utterances to deal with recognition errors. An overview of our strategy is shown in Figure 4.2.

4.3.1 Making Effective Confirmation

Confidence Measure (CM) is useful in selecting reliable candidates and controlling confirmation strategy. By setting two thresholds $\theta_1, \theta_2 (\theta_1 > \theta_2)$ on content-word CM (CM_w), we adopt the confirmation strategy as follows.

1. $CM_w > \theta_1$ $\rightarrow$ accept the hypothesis

2. $\theta_1 \geq CM_w > \theta_2$ $\rightarrow$ make confirmation to the user
   "Did you say ...?"

3. $\theta_2 \geq CM_w$ $\rightarrow$ reject the hypothesis

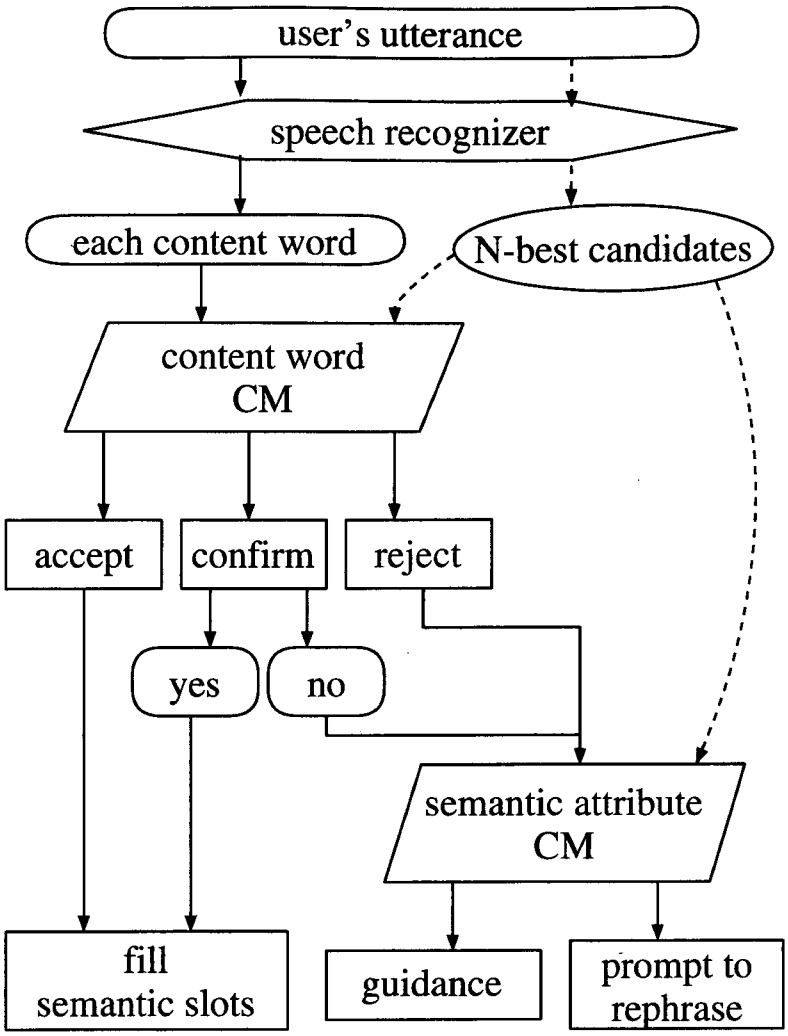


Figure 4.2: Overview of confirmation strategy

The threshold θ_1 is used to judge whether the hypothesis is accepted or should be confirmed, and the threshold θ_2 is used to judge whether it is rejected.

Because CM_w is defined for every content word, judgment among acceptance, confirmation and rejection is made for every content word when one utterance contains several content words. Suppose in a single utterance, one word has CM_w between θ_1 and θ_2 and the other has below θ_2, the former is given to confirmation process, and the latter is rejected. Only if all content words are rejected, the system will prompt the user to utter again. By accepting apparently correct words and rejecting unreliable candidates, this strategy focuses on only indistinct candidates and avoids redundant confirmation. These thresholds θ_1, θ_2 are optimized considering false acceptance (FA) and false rejection (FR)

using real development data.

Moreover, the system should confirm using task-level knowledge. It is not usual that users change the already specified slot values. Thus, recognition results that overwrite filled slots are likely to be errors, even though its CM_w is high. By making confirmation in such a situation, it is expected that false acceptance (FA) is suppressed.

4.3.2 Generating System-Initiated Guidance

It is necessary to guide the users to recover from recognition errors. Especially for novice users, it is often effective to instruct acceptable slots of the system. It will be helpful that the system generates guidance about the acceptable slots when the user is silent without carrying out the dialogue.

The system-initiated guidance is effective when speech recognition does not go well. Even when any successful output of content words is not obtained, the system can generate effective guidance based on the semantic attribute with high confidence. An example is shown in Figure 4.3. In this example, all the 10-best candidates are concerning a name of place but their CM_w values are lower than the threshold (θ_2), As a result, no word will be accepted nor confirmed. In this case, rather than rejecting the whole sentence and telling the user "Please say again", it is better to guide the user based on the attribute having high CM_c, such as "Which city is your destination?" This guidance enables the system to narrow down the vocabulary of the next user's utterance and to reduce the recognition difficulty. It will consequently lead next user's utterance to successful interpretation.

When recognition of a content word fails repeatedly in spite of high semantic-attribute CM, it is reasoned that the content word may be out-of-vocabulary. In such a case, the system should change the question. For example, if an utterance contains an out-of-vocabulary word and its semantic-attribute is inferred as "location", the system can make guidance, "Please specify with a name of prefecture", which will lead the next user's utterance into the system's vocabulary.

4.4 Experimental Evaluation

4.4.1 Task and Data

The strategy is evaluated on the hotel query task. We collected 120 minutes speech data by 24 novice users by using the prototype system described in chapter 3 (Figure 3.1)

utterance: "*shozai ga oosakafu no yado*"
(hotels located in Osaka pref.)
correct: Osaka-pref.@location

i	recognition candidates (<g>: filler model)
1	*shozai ga potoairando no* <g> located in Port-island
2	*shozai ga potoairando no* <g> located in Port-island
3	*shozai ga oosakafu no* <g> located in Osaka-pref.
4	*shozai ga oosakafu no* <g> located in Osaka-pref.
5	*shozai ga oosakashi no* <g> located in Osaka-city
6	*shozai ga oosakashi no* <g> located in Osaka-city
7	*shozai ga okazaki no* <g> located in Okazaki
8	*shozai ga okazaki no* <g> located in Okazaki
9	*shozai ga oohara no* <g> located in Ohara
10	*shozai ga oohara no* <g> located in Ohara

CM_w	content words
0.38	Port-island@location
0.30	Osaka-pref.@location
0.13	Osaka-city@location
0.11	Okazaki@location
0.08	Ohara@location

CM_c	semantic attributes
1	location

Figure 4.3: Example of high semantic attribute confidence in spite of low word confidence

Table 4.1: Distribution of CM_w (hotel task)

CM_w	# of outputs	# of correct answers	precision (%)
0.0 - 0.1	158	2	1.3
0.1 - 0.2	39	2	5.1
0.2 - 0.3	25	2	8.0
0.3 - 0.4	24	1	4.2
0.4 - 0.5	20	6	30.0
0.5 - 0.6	29	10	34.5
0.6 - 0.7	20	9	45.0
0.7 - 0.8	27	13	48.1
0.8 - 0.9	39	19	48.7
0.9 - 1.0	137	110	80.3
1.0	530	455	85.8
total	1048	629	60.0

[41]. The users were given simple instruction beforehand on the system's task, retrievable items, how to cancel input values and so on. The data is segmented into 705 utterances with a pause of 1.25 seconds. The vocabulary of the system contains 982 words, and the number of database records is 2040. A gender-dependent triphone model (3000 states, 16 mixtures) is used as the acoustic model [42].

Out of 705 utterances, 124 utterances (17.6%) are beyond the system's capability, namely they are out-of-vocabulary, out-of-grammar, out-of-task, or fragment of utterance. In the following experiments, we evaluate the system performance using all data including these unacceptable utterances in order to evaluate how the system can reject unexpected utterances appropriately as well as accept regular utterances correctly.

4.4.2 Optimization of Thresholds to Make Confirmation

The distribution of the content-word CM (CM_w) is shown in Table 4.1 for the collected data in the hotel query task. The result shows that high precision (# of correct answers / # of outputs) can be achieved for the range where CM_w is high. This means that proposing CM_w is an adequate criterion that represents speech recognition performance.

In section 4.3.1, we presented confirmation strategy by setting two thresholds θ_1, θ_2 for the content-word CM (CM_w). We optimize two threshold values that are needed for the confirmation strategy using the collected data. We count errors not by the utterance but by the content-word (slot). The number of slots to be filled is 804. In the experiment, we

use all collected data to decide two thresholds θ_1, θ_2. This is because it can be regarded that the threshold values do not change a lot between an open test and a closed test, since the distribution of CM is mainly dependent on the grammar and the acoustic model, which are common through the experiments, and the interval of the two thresholds is not so fine as 0.1. Therefore, equivalent performance can be expected for new data as long as the same grammar and same acoustic model are used.

Decision of the threshold θ_1

The threshold θ_1 decides between acceptance and confirmation. The value of θ_1 should be determined considering both the ratio of incorrectly accepting recognition errors (False Acceptance: FA) and the ratio of slots that are not filled with correct values (Slot Error: SErr). Namely, FA and SErr are defined as the complements of precision and recall rate of the output, respectively.

$$FA = \frac{\#\text{ of incorrectly accepted words}}{\#\text{ of accepted words}}$$

$$SErr = 1 - \frac{\#\text{ of correctly accepted words}}{\#\text{ of all correct words}}$$

We weight the FA because accepting an error damages the dialogue worse than rejecting a correct hypothesis. By minimizing this weighted loss function (wFA+SErr), we derive a value of θ_1 as 0.9 (see Figure 4.4).

Decision of the threshold θ_2

Similarly, the threshold θ_2 decides between confirmation and rejection. The value of θ_2 should be decided considering both the ratio of incorrectly rejecting content words (False Rejection: FR) and the ratio of accepting recognition errors into the confirmation process (conditional False Acceptance: cFA).

$$FR = \frac{\#\text{ of incorrectly rejected words}}{\#\text{ of all rejected words}}$$

If we set the threshold θ_2 lower, FR decreases and correspondingly cFA increases, which means that more candidates are obtained but more confirmation are needed. By minimizing FR+cFA, we derive a value of θ_2 as 0.6 (see Figure 4.5).

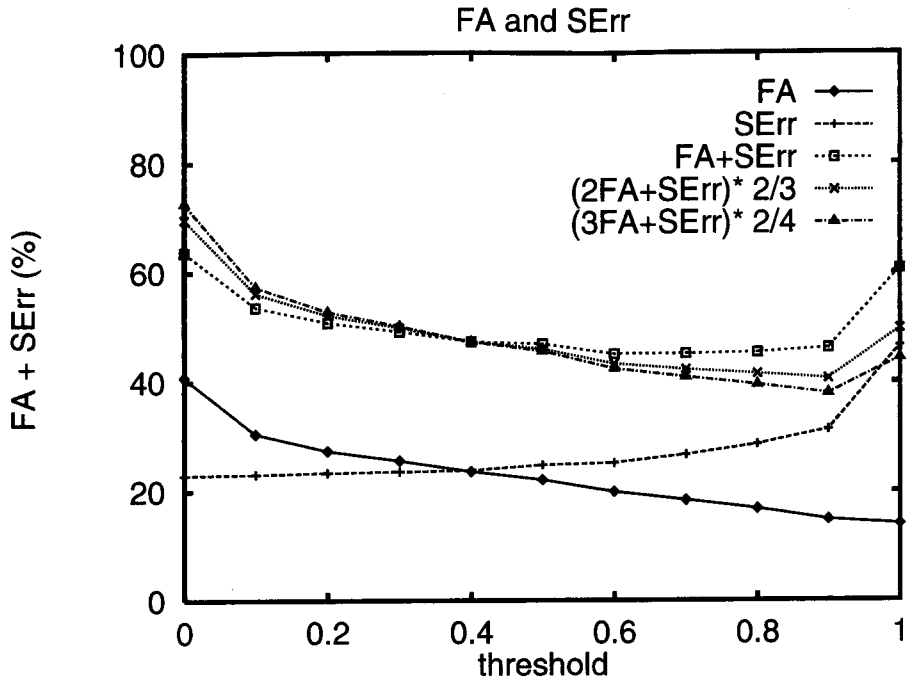


Figure 4.4: Operation curve of FA+SErr against threshold θ_1 (hotel task)

Table 4.2: Semantic accuracy compared with conventional methods (hotel task)

	FA+SErr	FA	SErr
only 1st candidate	48.6	24.7	23.9
no confirmation	44.1	19.5	24.6
with confirmation	39.9	15.3	24.6

FA: ratio of incorrectly accepting recognition errors
SErr: ratio of slots that are not filled with correct values

4.4.3 Comparison with Conventional Methods

In many conventional spoken dialogue systems, only 1-best candidate of a speech recognizer output is used in the subsequent processing. We compare our method with the conventional method that uses only 1-best candidate (Table 4.2).

In the 'no confirmation' strategy, the hypotheses are classified by a single threshold (θ) into either accepted or rejected. Namely, content words having CM_w over threshold θ are accepted, and otherwise simply rejected. In this case, a threshold value of θ is set to 0.6 that makes the semantic error rate (FA+SErr) minimal. In the 'with confirmation' strategy, the proposed confirmation strategy is adopted using θ_1 and θ_2. We set $\theta_1 = 0.9$

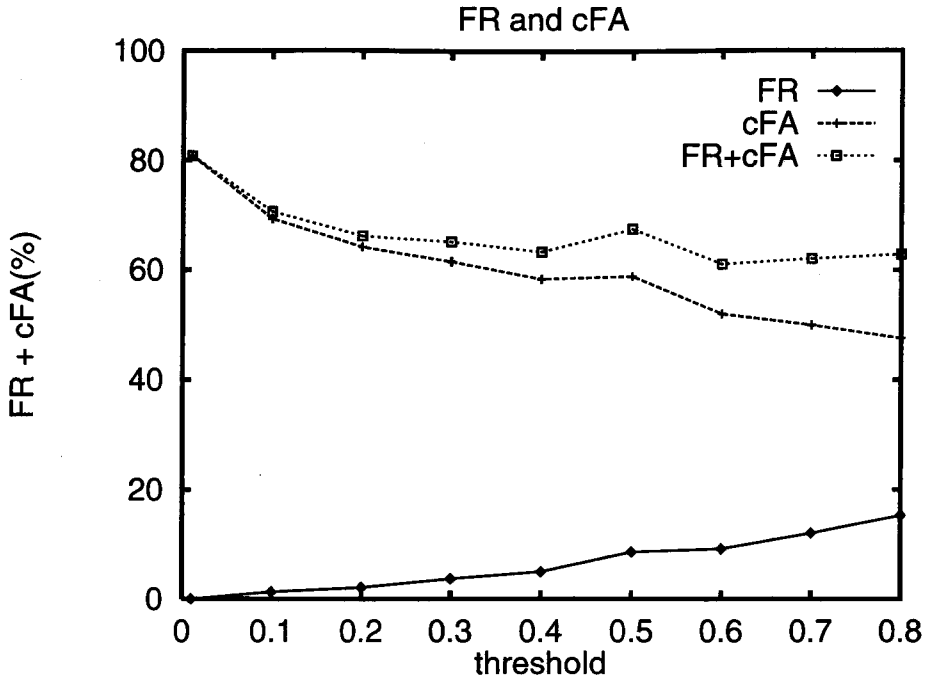


Figure 4.5: Operation curve of FR+cFA against threshold θ_2 (hotel task)

and $\theta_2 = 0.6$. 'FA+SErr' in Table 4.2 means FA(θ_1)+SErr(θ_2), on the assumption that the confirmed phrases are correctly accepted or rejected. We regard this assumption as appropriate, because users tend to answer 'yes' simply to express their affirmation [43], so the system can distinguish affirmative answer and negative one by recognizing simple 'yes' utterances correctly.

As shown in Table 4.2, the semantic error rate (FA+SErr) is reduced by 4.5% by the 'no confirmation' strategy compared with the conventional method. And 'with confirmation' strategy, we achieve 8.7% improvement in total. This result proves that our method successfully eliminates recognition errors.

By making confirmation, the interaction becomes robust, but accordingly the number of utterances in the dialogue increases. If all candidates having CM_w under θ_1 are given to confirmation process without setting θ_2, 332 vain confirmation for incorrect contents are generated out of 400 candidates, as shown in Table 4.3. By setting θ_2, 102 candidates having CM_w between θ_1 and θ_2 are confirmed, and the number of incorrect confirmation is suppressed to 53. Namely, the ratio of correct hypotheses and incorrect ones being confirmed are almost equal. This result shows only indistinct candidates are given to confirmation process whereas unreliable candidates are rejected.

Table 4.3: Effect of setting θ_2 (hotel task)

	# of outputs	# of correct hypotheses	precision
$\theta_1 > CM_w > 0$	400	68	17%
$\theta_1 > CM_w \geq \theta_2$	102	49	48%

(Here, $\theta_1 = 0.9$, $\theta_2 = 0.6$)

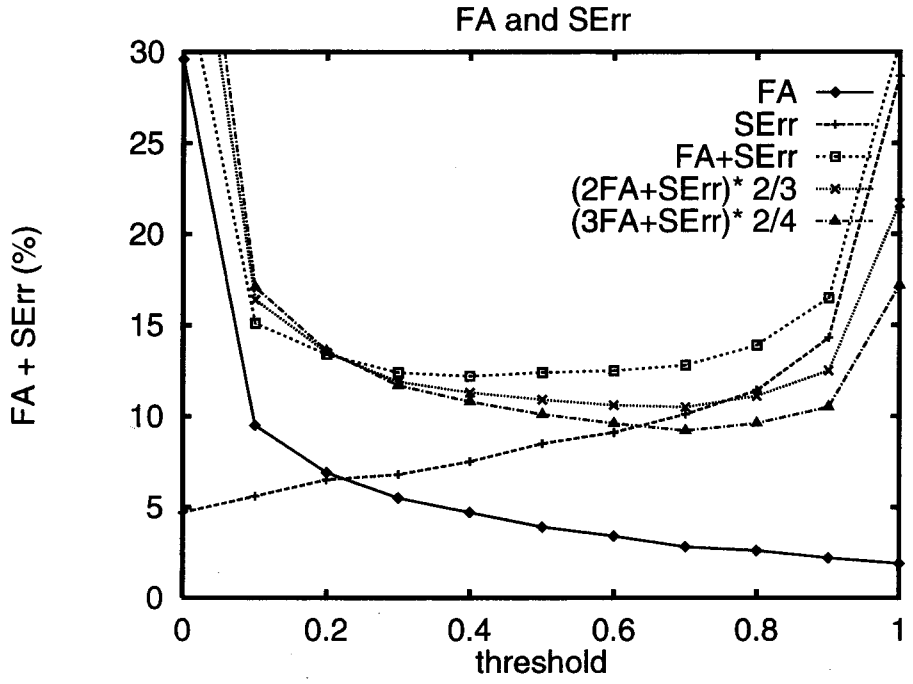


Figure 4.6: Operation curve of FA+SErr against threshold θ_1 (ATIS task)

4.4.4 Evaluation on Other Tasks

The method is applied to the tasks of ATIS [44] and DARPA Communicator [45] in order to verify the effectiveness and generality.

First, we evaluated on the ATIS-3 database. Speech recognition results were obtained with the Bell Labs. system, and the stochastic and automatically trained FSM (finite state machine) parser is applied to extract semantic slots [46]. We used first 669 utterances as the test-set. The vocabulary size is 1047 and there are a few out-of-vocabulary words in the test-set. Since the matched stochastic acoustic and language models trained with a large scale of speech and text database are available on the ATIS task, recognition accuracy is much better than the previous task.

In this case, the best operating point of the loss functions is shifted to smaller values

Table 4.4: Semantic accuracy compared with conventional methods (ATIS)

	FA+SErr	FA(%)	SErr(%)
only 1st candidate	10.9	3.7	7.3
no confirmation	11.1	4.1	7.0
with confirmation	9.2	2.2	7.0

Table 4.5: Semantic accuracy compared with conventional methods (Communicator)

	FA+SErr	FA(%)	SErr(%)
only 1st candidate	39.7	19.4	20.3
no confirmation	36.8	15.1	21.7
with confirmation	34.0	12.3	21.7

as in Figure 4.6, and we have derived the thresholds as θ_1=0.7 and θ_2=0.4. Semantic error rates are listed in Table 4.4. On this task, selection of hypotheses with the confidence measure reduces false acceptance, but increases slot errors much. Combined with the confirmation strategy, however, we could reduce total errors by 16% relative.

We also applied the method to the DARPA communicator system developed at Bell Labs. At the moment, 1395 utterances were collected via telephone lines with a prototype system. We derive the thresholds as θ_1=0.8 and θ_2=0.6. On this task, too, consistent improvement is confirmed with the proposed framework as shown in Table 4.5.

4.4.5 Effectiveness of Semantic-Attribute CM

In Figure 4.7, the performance of content-word CM and semantic-attribute CM is shown for the hotel query task. They are evaluated by the weighted sum of '3FA+SErr'. It is observed that semantic-attribute CM is estimated more correctly than content-word CM. This fact suggests that the semantic attribute can be estimated correctly even when successful interpretation is not obtained from the content-word CM.

In the test data, there are 148 slots[1] that are not obtained correctly by the content-word CM. It is also observed that 52% of semantic attributes with CM_c over 0.9 are correct. Such slots amount to 34. Namely, our system can generate effective guidance for 23% (34/148) of utterances that had been only rejected in conventional methods. For

[1]Out-of-vocabulary and out-of-grammar utterances are included .

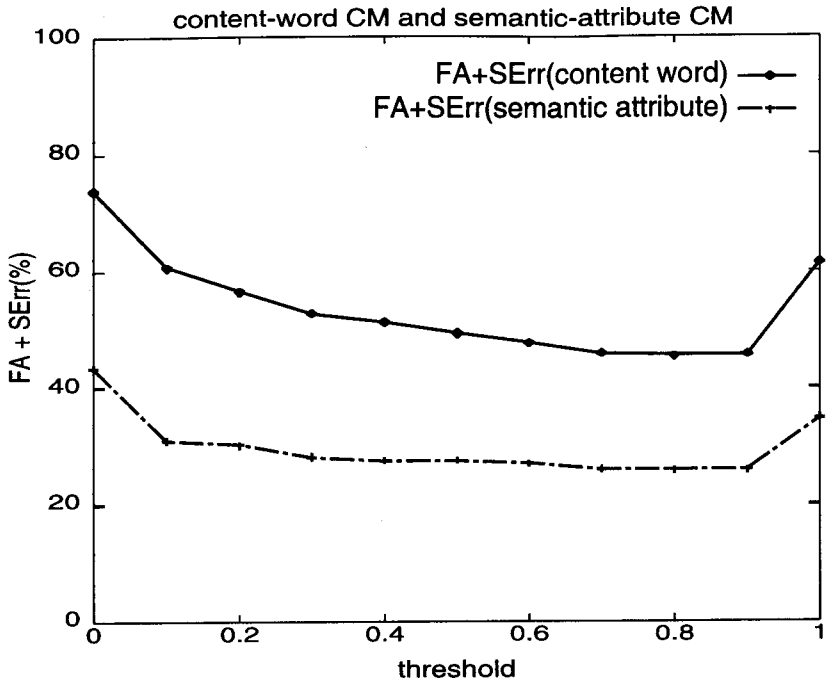


Figure 4.7: Performance of word CM and category CM (hotel task)

those slots with $CM_c = 1$, we can generate guidance with accuracy of 90% (9/10). And by making confirmation for the slots having CM ($1.0 > CM_c \geq 0.5$) like "Are you saying about price?", guidance is generated for 16% (24/148) utterances that had been only rejected in conventional methods.

As for ATIS data, we could recover the concept for 40 out of 66 slots (61%) that are not filled with the content-word CM. Some of them are incomplete, for example, simple `TIME` concept instead of `FROM_TIME` that specifies the departure time. But the information is useful when combined with the dialogue management.

4.5 Conclusions

We present dialogue management using two concept-level CMs in order to realize robust interaction. The content-word CM provides a criterion to decide whether an interpretation should be accepted, confirmed or rejected. This strategy is realized by setting two thresholds that are optimized balancing false acceptance and false rejection. The interpretation error (FA+SErr) is reduced by 4.5% with no confirmation and by 8.7% with confirmation. Moreover, we define CM on semantic attributes, and propose a new method

to generate effective guidance. The concept-based CM realizes flexible dialogue management in which the system can make effective confirmation and guidance by estimating the user's intention.

Chapter 5

Generating Guiding Questions to Constrain Information Retrieval Results using Structure of Domain Knowledge

5.1 Introduction

In the past years, a great number of spoken dialogue systems have been developed. Their typical task domains include airline information [47, 45, 29] and train information [48, 40, 31, 30]. Most of them model speech understanding process as converting recognition results into semantic representations equivalent to database query (SQL) commands, and dialogue process as disambiguating their unfixed slots. Usually, the semantic slots are defined a priori and manually. The approach is workable only when data structure of the application is well-organized typically as a relational database (RDB).

Different and more flexible approach is needed for spoken dialogue interfaces to access information described in less rigid format, in particular normal text database. For the purpose, information retrieval (IR) technique is useful to find a list of matching documents from the input query. Typically, keywords are extracted from the query and statistical matching is performed. Call routing task [49] can be regarded as the special case.

In IR systems, many candidates are usually obtained as a query result, thus there is a significant problem of how to find the user's intended item among them. Especially, either on the telephone or electrical appliances, there is not a large screen displaying the candidates, and all the query results cannot be presented to a user. So it is desirable for the system to narrow down the query results interactively. Moreover, interactive query is

friendlier to novice users rather than requiring them to input a detailed query from the beginning.

In this chapter, we address a dialogue strategy to find the user's intended item from the retrieved result, which is initiated by a spontaneous query utterance. We first present the construction of statistical language model for information query tasks in section 5.2. For information query tasks where semantic slots are not assumed, statistical language models based class N-gram model are effective. We construct the domain-adapted language model by merging a task-specific model made with a small set of sentences in the target domain with a general one trained with a large amount of test data. In section 5.3, we describe a method to generate a guiding question that narrows down the query results efficiently, using an example of the restaurant query task. The question is selected based on an information theoretic criterion. In section 5.4, we present a dialogue management method for a query task on the appliance manual where structured task knowledge is available. We propose a confirmation strategy by making use of a tree structure of the manual, and define three cost functions for selecting question nodes. The method is evaluated by the number of average dialogue turns.

Although there are previous studies on optimizing dialogue strategies [5, 50, 51], most of them assume the tasks of filling semantic slots that are definitely and manually defined, and few focus on follow-up dialogue of information retrieval. For example, Denecke proposed a method to generate guiding questions by making use of a tree structure constructed by unifying retrieved items based on semantic slots [52]. In this chapter, we do not assume any structure of semantic slots. Instead, we make use of distribution of document statistics or a structure of task knowledge. We also investigate cost functions for optimal dialogue control by taking into account of speech recognition errors.

5.2 Statistical Language Model for Information Query Tasks

It is important that the vocabulary and expressions that are specific to the target domain are covered in speech interfaces. So, syntactic constraint and semantic interpretation rules are often described as grammar rules in conventional systems. However, speech recognition and semantic interpretation based on such grammar rules are usually so rigid especially when users are allowed to speak freely [53]. On the other hand, statistical

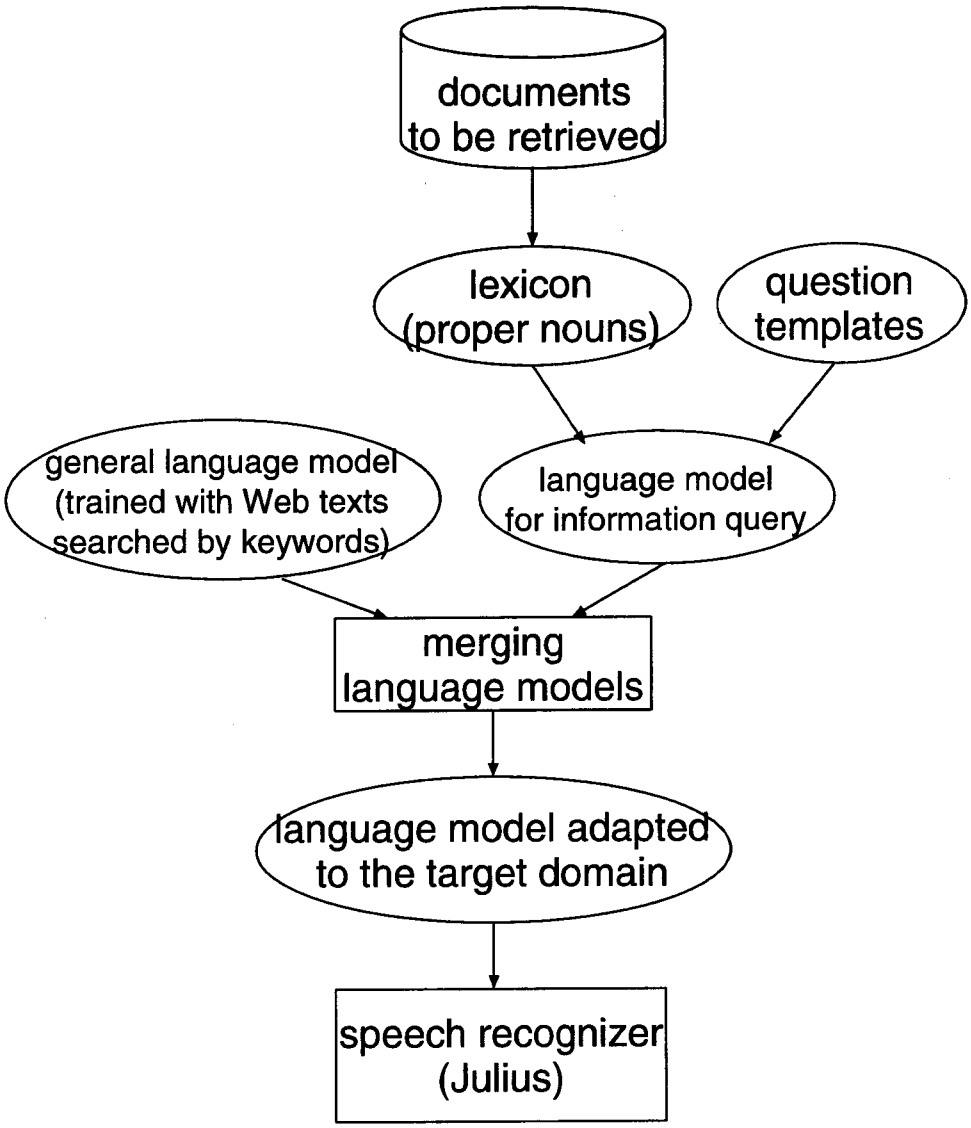


Figure 5.1: Flow of constructing statistical language models for information query

methods based on large corpora have been successful. Although various utterances can be recognized flexibly with statistical language models, it is usually difficult to collect sufficient amount of training data for every domain. Moreover, expressions specific to the domain such as the name of places are not covered with a general statistical language model, such as a language model trained with newspapers. In the system that directly uses speech recognition results as an input of the search engine of WWW such as [54], sufficient coverage and recognition accuracy cannot be expected unless the language model is adapted to the target domain.

In this section, we present a method to construct the flexible language model adapted

<RESTAURANT> *wa doko desu ka?*
(Where is <RESTAURANT>?)

<RESTAURANT> *no denwa bangou wo oshiete kudasai.*
(Please tell me the telephone number of <RESTAURANT>.)

<RESTAURANT> *no eigyo jikan wo oshiete kudasai.*
(Please tell me the business hours of <RESTAURANT>.)

<DISTRICT> *no* <RESTAURANT> *wa doko desuka?*
(Where is <RESTAURANT> in <DISTRICT>?)

<RESTAURANT> *ni* <TRANSPORT> *de ikitai no desuga.*
(I want to go to <RESTAURANT> by <TRANSPORT>.)

<PLACE> *de ichiban yuumei na* <FOOD_TYPE> *no mise wa nan desuka?*
(What is the most famous restaurant that serves <FOOD_TYPE> in <PLACE>?)

oishii <SNACK> *wa doko de taberareru no desuka?*
(Where can I eat delicious <SNACK>?)

<PLACE> *no chikaku no oishii* <REST_TYPE> *wo oshiete kudasai.*
(Please tell me the delicious <REST_TYPE> near <PLACE>.)

Figure 5.2: Examples of query templates

to the target domain by mixing two statistical language models: one is a general model trained with a large amount of text data, the other is a task-specific model made with a small set of sentences in the target domain. The flow of processes is shown in Figure 5.1. This model makes it possible to recognize various utterances flexibly while covering vocabulary and expressions specific to the domain.

We use sentences written by hand for restaurant query of the Tokyo area as the target domain texts. These sentences consist of 335 templates as shown in Figure 5.2 and 131 questions that corresponds to specific restaurants such as "Is there any restaurants that can be reserved by the Internet?" We generate a class N-gram model using these templates and questions. The number of the defined classes is seven as shown in Figure 5.3.

By registering proper nouns corresponding to the classes into the word dictionary of the obtained class N-gram model, the language model of the restaurant query task is composed. The vocabulary size is 925. The language model constructed here is a pseudo class N-gram model, since the occurrence probability of all words in classes is regarded as

<RESTAURANT> : name of restaurants (245)

<FOOD_TYPE> : type of food (68)

- *chugoku ryori* (Chinese), *nihon ryori* (Japanese), *itaria ryori* (Italian), ...

<PLACE> : name of places (125)

- *Ebisu, Shinjuku, Hibiya, Yurakucho*, ...

<REST_TYPE> : type of restaurants (7)

- *izakaya* (tavern), *kissaten* (coffee house),
  ramen-ya (Chinese noodle shop), ...

<SNACK> : the name of food (129)

- *okonomiyaki, paeria* (paella), *kaiten-sushi*, ...

<TRANSPORT> : transportation (5)

- taxi, trains, bus, *JR*, subways

<district> : name of districts (4)

- *Chuo-ku, Minato-ku, Shibuya-ku, Shinjuku-ku*

Number in () denotes the number of instances in the class.

Figure 5.3: Classes of proper nouns and their instances

1.

The constructed language model for restaurant query is merged [55] with more general language model (the vocabulary size is 19447) that covers the gourmet-recipe domain and trained with Web texts collected by a search engine [56]. The vocabulary size of the merged language model is 20370. Although the language model in the gourmet-recipe domain is trained with a large amount of texts and covers various utterances flexibly, it does not contain proper nouns and expressions specific to restaurant query tasks. By merging with the language model constructed from target-domain sentences, we can get a flexible language model that covers both proper nouns and wide variety of expressions.

Restaurant A
Chinese noodles, meat dumpling, Shinjuku, Kabukicho

Restaurant B
Chinese noodles, meat dumpling, Chinese tea and snacks, Shinjuku, Kabukicho

Restaurant C
Chinese noodles, meat dumpling, noodles with boiled-pork-ribs, Takadanobaba

Restaurant D
Chinese noodles, fried garlic, Ebisu

...

Figure 5.4: Example of domain knowledge

5.3 Dialogue Strategy in General Information Query Tasks

Interaction in an information query task can be regarded as a process seeking a common part between the user's request and system knowledge. In order to help users find their intended items from the system knowledge, the system has to carry out not only interpreting what users say but also showing the relevant portion of the system knowledge.

We assume that users can freely specify and retract query keys based on their preference to information query systems. If many candidates still remain even after specifying all possible their preference to the system, the user may have difficulty in narrowing down further the query result. Thus, the system should generate efficient guiding questions to help users find their intended items.

In this section, we assume the system knowledge as a pair of an item and a set of keywords (Figure 5.4). We define keywords as a set of words representing contents of the items, and their categories such as place and food are given. This is similar to indexing words in a conventional information retrieval task. Note that it is not needed that the system knowledge is structured like an RDB.

Keywords are extracted from a user's utterance, and are matched with the system knowledge. Here, we adopt the following matching function for each item j.

$$L_j = \sum_{i \in K_j} \left(CM_i * \log \frac{N}{df_i} \right)$$

Here, K_j is a set of keywords for item j. CM_i is a confidence measure of speech recognition for keyword i [57], N is the total number of items, and df_i is the number of items including keyword i. Intuitively, keyword that is recognized with high confidence and does not appear in many items gets higher likelihood L_j by CM_i and df_i, respectively.

Then, we define amount of information that is obtained when the system generates yes/no question and the user answers it. Here, C is a current query condition, A is a condition that is added by the system's question, and $count(x)$ is the number of items that satisfy the condition x. The condition consists of the conjunction of the keywords the user has specified. Suppose each item occurs by equal likelihood, the following equation denotes the likelihood $p'(A_{yes})$ that the yes/no question corresponding to the adding condition A will be answered as "yes".

$$p'(A_{yes}) = \frac{count(C \cap A)}{count(C)}$$

We weight each item j with the likelihood L_j.

$$p(A_{yes}) = \frac{\sum_{j \in \{C \cap A\}} L_j}{\sum_{j \in \{C\}} L_j}$$

The amount of information that is obtained when the user's answer is "yes" is represented as follows.

$$I(A_{yes}) = \log_2 \frac{1}{p(A_{yes})}$$

The following equation gives $H(A)$, the expected value of amount of information that is obtained by generating a question about condition A and getting user's answer ("yes" or "no").

$$H(A) = \sum_{x \in \{yes, no\}} p(A_x) \log_2 \frac{1}{p(A_x)}$$

By calculating $H(A)$ for all conditions A that can be added to the current query condition, the system generates the question that has the maximum value of $H(A)$. The question is generated using the category information of the keyword.

Because the condition A is selected by a viewpoint of narrowing down the current set of items efficiently, the selected condition may be unimportant for the user. In such a case, it is not cooperative to force the user an affirmative or negative reply. Our system does not force the reluctant decision by allowing the user to say "It does not matter anyhow." Instead, the system presents the second best proposal.

We explain the method with the following example in our restaurant query system in the Tokyo area. When a user says, "Please tell me a restaurant where I can eat Chinese noodle and meat dumpling in Shinjuku area." three keywords are extracted: "Shinjuku", "Chinese noodle" and "meat dumpling". As a result of the matching using these three keywords, 11 query results are obtained. It is not user-friendly to read out all of the 11 query results with a TTS (text-to-speech) system. Here, the expected value of amount of information $H(A)$ is calculated for each condition that corresponds to keywords included in the matched items except for the three keywords, "Shinjuku", "Chinese noodle" and "meat dumpling". Then, we select the keyword "noodles with boiled-pork-ribs" that has the maximum value $H(A)$. By generating a question like "Would you like one which serves noodles with boiled-pork-ribs?" and obtaining a reply from the user, the system adds the new condition and narrows down the candidates efficiently. If the user thinks that the condition "noodles with boiled-pork-ribs" is not important and tells the system so (for example "Either will do."), the system can show the second best proposal, "Would you like one located in Kabukicho area?" Thus, the query result can be narrowed down without forcing the user unnatural yes/no answers.

5.4 Dialogue Strategy for Query on Appliance Manuals

In this section, we present another efficient solution in the case that the structure or hierarchy of task knowledge is available. The task here is to find the appropriate item in the manual of electric appliances with a spoken dialogue interface. Such an interface will be useful as the recent appliances become complex with many features and so are their manuals. In the appliances such as VTR (Video Tape Recorder) and FAX machines, there is not a large screen to display the list of matched candidates to be selected by the user. Therefore, we address a spoken dialogue strategy to determine the most appropriate one from the list of candidates.

An alternative system design is the use of directory search, as adopted in voice portal systems, where the documents are hierarchically structured and the system prompts users to select one of the menu from the top to the leaf. The method is rigid and not user-friendly since users often have trouble in selection and want to specify by their own expression. The proposed system allows users to make queries spontaneously and makes use of the

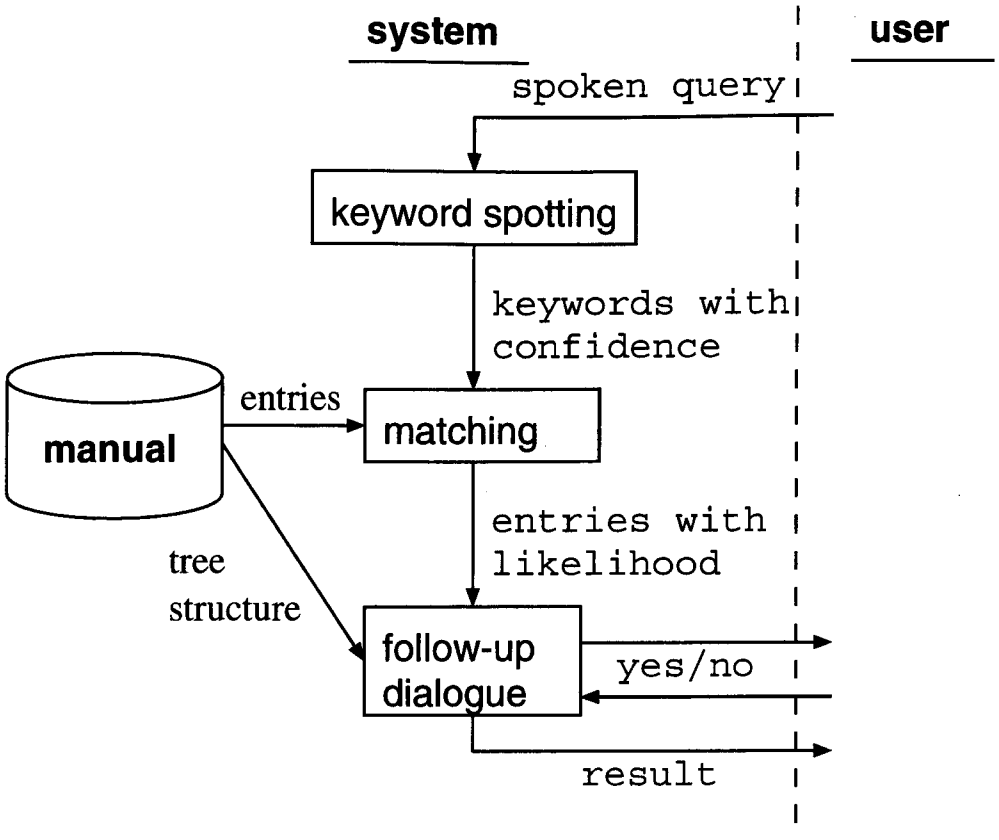


Figure 5.5: Overview of interactive manual query system

directory structure in the follow-up dialogue to determine the most appropriate one.

5.4.1 System Overview

An overview of the system is illustrated in Figure 5.5. It consists of following processes.

1. Keyword spotting from user utterances using an ASR (automatic speech recognition) system [28]

   A natural spoken language query is accepted and keywords are extracted. A confidence measure CM_i is assigned to each keyword i based on the N-best recognition result [57].

2. Matching with manual items (documents)

   The extracted keywords are matched with a set of manual items. The matching is performed on the initial portion (index and first summary paragraph) of each

manual section. We adopt the following matching score function for an item j. K_j is a set of keywords for item j.

$$L_j = \frac{1}{n_j} \sum_{i \in K_j} (CM_i * \log \frac{N}{df_i})$$

Here, df_i is the number of items that contain keyword i referred as a document frequency and N is the total number of items. The inverse document frequency (idf) is weighted with a confidence measure CM_i and summed over keywords, then normalized by n_j, the number of keywords in the item j.

3. Generating dialogue to determine the most appropriate one from the list of candidates

   As a result of the matching, many candidates are usually found. They may include irrelevant ones because of speech recognition errors. But it is not practical to read out all of them in order with a TTS (text-to-speech) system. Therefore, dialogue is invoked to narrow down to the intended one. This dialogue is restricted to system-initiated "yes/no" questions in order to avoid further recognition errors and back-up dialogue. The dialogue strategy is explained in the next subsection.

5.4.2 Dialogue Strategy using Structure of Manual

If one of the candidates is more plausible than others with a significant margin, we should make confirmation on it. When there are many candidates with similar confidence and they can be hierarchically grouped into several categories, we had better first identify which category the intended one belongs to. In this study, we make use of the section structure of the manual, i.e. section is the first layer, sub-section is the second-layer, and so on. The tree structure is automatically derived from its table of contents. An example for VTR manual is shown in Figure 5.6.

For each node of the tree, likelihood L'_j is assigned as follows.

- For a leaf node, the matching score L_j is assigned after normalizing so that the sum over all leaves (manual items) is 1.0.
- For a non-leaf node, the sum of the likelihood of its children nodes is assigned.

Then, a dialogue is generated as follows.

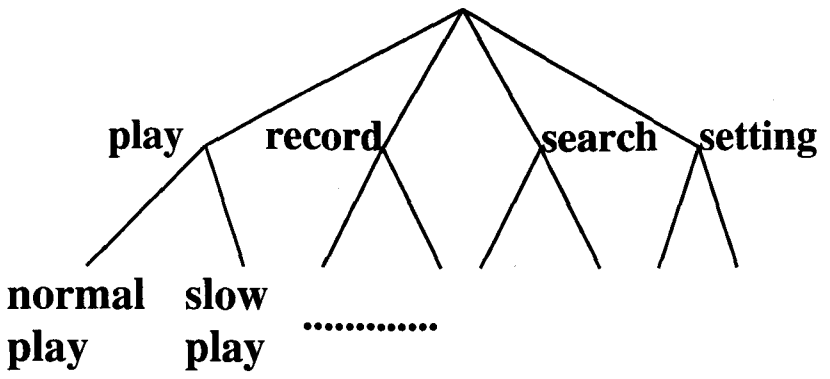


Figure 5.6: Example of tree structure of manual

1. Among ancestor nodes of the leaf of the largest likelihood L'_j, pick up the one whose heuristic cost function described below is smallest.

2. Make a "yes/no" question on the node, for example "Do you want to know about...?" The content of the question is associated with the section title.

3. If the user's answer is "yes", eliminate the nodes other than descendants of the confirmed node. If the answer is "no", eliminate all descendants of the denied node.

4. Repeat the process until only one node (or less than a threshold α) remains.

The above processes are illustrated in Figure 5.7.

We define following three heuristic cost functions in order to realize an efficient dialogue.

- $h_1(j) = |L'_j - 0.5|$

  This makes a question on the most ambiguous node whose likelihood L'_j interpreted as a posteriori probability is close to 0.5.

- $h_2(j) = L'_j * Node_j(yes) + (1 - L'_j) * Node_j(no)$

  Here, $Node_j$ is the number of remaining nodes when the answer is "yes" or "no". This function takes the approximate number of following questions into account.

- $h_3(j) = L'_j * Ques_j(yes) + (1 - L'_j) * Ques_j(no) + 1$

  $Ques_j$ is the estimated number of subsequent questions needed when the answer is "yes" or "no". It is computed recursively by expanding the sub-tree, and is assigned

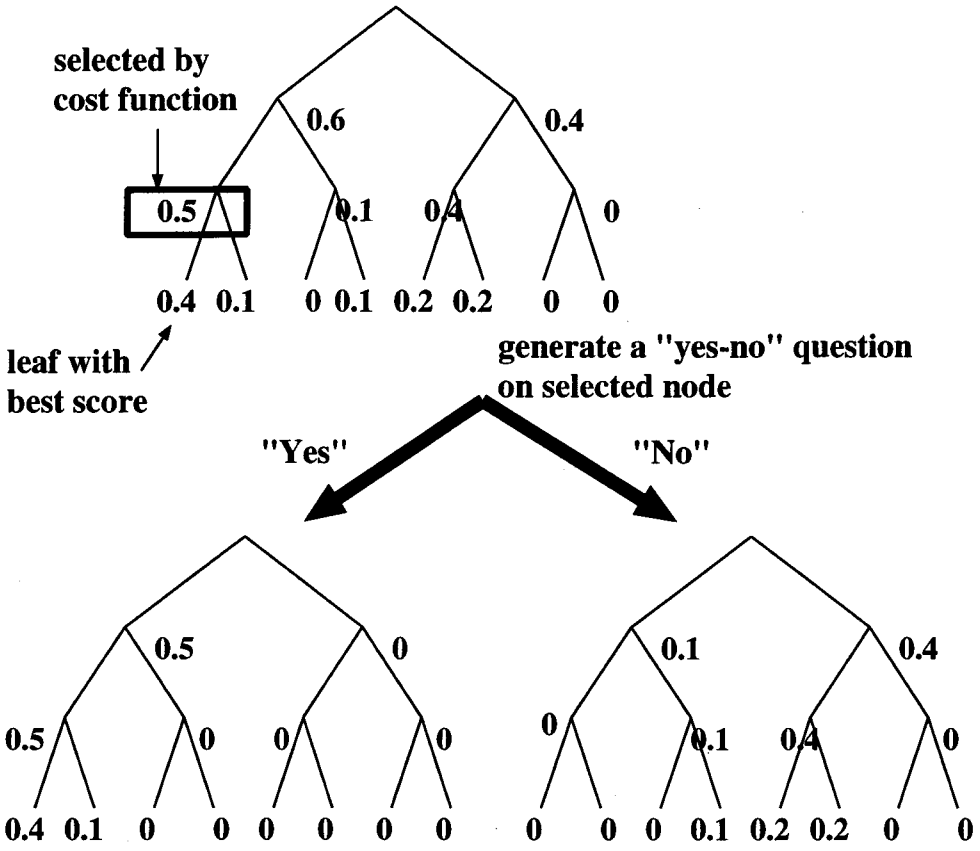


Figure 5.7: Use of manual structure and cost function for dialogue control

with 0 when the number of remaining nodes is under a threshold (α). This α means the number of candidates that can be presented to users. Here, we set $\alpha = 3$.

These are experimentally compared in the next subsection.

5.4.3 Experimental Evaluation

Task and System Implementation

The proposed system is implemented for the query task on a VTR manual that consists of 111 pages and 47 items. The derived tree structure is of three levels. The number of keywords used for matching is 137.

The speech recognition system is based on our large vocabulary continuous speech recognition engine. The language model is initially based on a finite state grammar and extended to combine statistical models derived from the domain-specific corpus [58], that is the manual text in this task. The acoustic model is a gender-dependent phonetic

Table 5.1: Evaluation result of our dialogue strategy with text input

# matched candidates	12.4		
query success rate	93%		
average rank of correct item (# turns by baseline)	3.2		
# turns by proposed cost functions	h_1	h_2	h_3
	2.4	2.5	2.8

tied-mixture (PTM) triphone model [59] trained with the 40-hour JNAS speech corpus.

For collecting evaluation data, we had 14 subjects and each made 10 queries on given scenarios (query sentences are not given), and several spontaneous queries without any scenarios. In total, we had 195 query utterances, of which 157 could be coped with the given manual, thus used as the test-set. Sample queries are "I want to change the recording reservation." and "Can I watch TV while recording another program?"

As for evaluation measures, we first compute the rate of query success where the correct manual item is contained in the candidate list by the initial matching. Then, the system is evaluated by the necessary dialogue turns equivalent to the number of questions before the correct item is identified. It is compared with the baseline case where the candidates are presented to the user in order of the matching score L_j and the number of dialogue turns is equivalent to the rank of the correct item.

Evaluation with Text Input

At first, the system is evaluated with text input, which is transcription of the collected queries. The result is shown in Table 5.1.

On the average, the matching result consists of 12.4 candidates and contains correct one for 93% of the tractable queries. The average rank of the correct item is 3.2, which means, if we make confirmation in order of the matching score L_j, we need 3.2 turns on the average. With dialogue based on the heuristic cost functions, it can be reduced to 2.4 (h_1), 2.5 (h_2) and 2.8 (h_3), respectively.

We have not yet identified the reason why performance by the apparently most accurate function h_3 is not good. We conjuncture that the difference of the cost functions does not matter so much in this framework as long as they are reasonable.

Table 5.2: Precision of keywords and their confidence measures

confidence measure of keyword	1	1 - 0.9	0.9 - 0.8	0.8 - 0.7	0.7 -	total
# correctly recognized words	279	15	10	18	16	338
# incorrectly recognized words	63	17	20	49	60	209
precision	82%	47%	33%	27%	21%	62%

Table 5.3: Evaluation result of our dialogue strategy with speech input

# matched candidates	13.3		
query success rate	87%		
average rank of correct item (# turns by baseline)	4.1		
# turns by proposed cost functions	h_1	h_2	h_3
	2.9	2.9	3.2

Evaluation with Speech Input

Next, we made experiments using the spoken queries and the speech recognition system. The distribution of recognized keywords and corresponding confidence measures is shown in Table 5.2. The precision for the keywords with high confidence measures is better, thus the confidence measure works well. Summary of the result is given in Table 5.3.

The average number of matched items is 13.3 and the success rate is 87%. Some degradation from the case of text input is observed. The average rank of the correct item is 4.1. For reference, if we do not use the confidence measure CM_i, the figure is 4.4, which verifies the effect of the confidence measure. The proposed dialogue strategy with either heuristic function reaches the correct one in around 3 turns, which is 30% reduction compared with the baseline.

It should be noticed that, although the initial matching accuracy is lowered with the speech input, the improvement by the proposed strategy is larger and the number of dialogue turns is close to the text-input case. The result confirms that the proposed framework is effective in speech interface.

5.5 Conclusions

We present a method to generate guiding questions for narrowing down users' query results obtained by an information retrieval system. By selecting the most efficient item, the dialogue is restricted to system-initiated "yes/no" questions. We have evaluated the method with a query task on the appliance manual where structured task knowledge is available. The number of average dialogue turns is reduced by about 30% compared with a baseline method in which the candidates are confirmed according to their matching scores. This result demonstrates that the proposed system helps users find their intended items more efficiently.

Chapter 6

Conclusions

We have studied methods to realize spoken dialogue systems with flexible dialogue strategies. We have addressed the following problems: the disambiguation in information retrieval with speech interfaces and domain-independent construction of language models and dialogue models. The ambiguity is included in speech and natural language by nature. This problem is solved through the dialogue in which efficient confirmation and effective guidance are generated appropriately. As spoken dialogue systems are used in wide and complicated domains, the dialogue strategies cannot be described by hand and must be based on domain-independent information. The construction of the systems and their language model should also be domain-independent, as the systems have been used in more various domains.

As the basis of domain-independent dialogue models, the annotated corpora are indispensable to train statistical methods. We implemented a program to infer the utterance-unit tag, which is one of discourse tags. The method uses the features of surface expressions at the end of sentences. The inference is achieved accuracy of 86% and 73% in closed and open test, respectively. The program is useful for annotation, and the framework to extract the features automatically can be extended to a more general-purpose inference programs.

We have also developed a domain-independent platform with a flexible key-phrase spotter based on combined language models. The key-phrase spotter realizes flexible speech understanding for various domains. The platform semi-automatically generates database query systems with speech interfaces by preparing a language model adapted to the target domain using the information extracted from the domain database. As the language model, we constructed the combined language model by integrating generated

grammar rules and statistical models. It introduces linguistic constraint for both domain-dependent key-phrase parts and domain-independent parts. The phrase spotter based on the combined language model extracts the attribute-value pair, which is needed in performing the task having the slot-type data structure. It improves the semantic accuracy by 15.5% compared with the conventional method decoding the whole sentence with a fixed grammar. The proposed method is useful for constructing spoken dialogue systems in various domains.

The most significant cause of the ambiguity in spoken dialogue systems is the speech recognition errors. We have proposed a flexible dialogue strategy to make confirmation and guidance in order to manage recognition errors. To avoid redundant confirmation, we calculate confidence measures for each content word and decide whether it should be confirmed. The threshold deciding whether to make confirmation is determined optimally considering the balance between acceptance and rejection. Moreover, the confidence measure is calculated also for semantic categories, enables the system to generate effective guidance even when any confident interpretation is not obtained by the content-word level. The dialogue strategy using the two-level confidence measures improves the interpretation accuracy by 8.7% in the hotel query task while suppressing redundant confirmation.

The ambiguity is also arisen by natural language expressions and a lot of query results. We have addressed how to generate effective guidance when many candidates are obtained as query results. The system should guess the user's intention and generate the most efficient question to accomplish the goal. We have proposed a method to generate such questions using a distribution of document statistics and a structure of task knowledge. The method does not assume the semantic slots conventionally used. When the hierarchical structure of a domain such as the table of contents of a manual is available, the optimal guiding question is generated by using the structure. The method is implemented in the VTR manual query task. The strategy reduces the number of average dialogue turns by about 30% compared with the baseline method in which the candidates are confirmed according to their matching scores.

The thesis has discussed methods to build flexible spoken dialogue systems in a domain-independent manner. We design the system not by describing the rule dependent on the target domain but by using domain-independent information and information derived from the domain database automatically.

The misunderstandings caused by speech recognition errors have been decreased by the

proposed confirmation strategy. As the strategy decides a response for a single utterance, a dialogue strategy covering two or more utterances is more desirable when the errors occur successively. We have also proposed the strategy to select an optimal candidate to narrow down a lot of query results derived by the ambiguity of natural language. To make the dialogue more user-friendly, a method to generate the surface expression of the guidance should be addressed in addition to the selection of the optimal content.

Acknowledgements

I would like to express my gratitude to Professor Hiroshi G. Okuno for supervising this thesis and for his suggestion and encouragement.

I also greatly appreciate Associate Professor Tatsuya Kawahara. I could not have accomplished this work without his continuous guidance and warm support.

I am grateful to Professor Katsumi Tanaka and Professor Toru Ishida for their invaluable comments to the thesis.

I would like to express my appreciation to Emeritus Professor Shuji Doshita and Associate Professor Masahiro Araki (currently at Kyoto Institute of Technology) for their enlightening guidance in the bachelor and master course.

I owe a great deal to the members of Professor Okuno's Laboratory (formerly Speech Media Laboratory and Professor Doshita's Laboratory). I had many suggestions from Dr. Akinobu Lee (currently at NAIST). I am also indebted to Mr. Katsuaki Tanaka, Mr. Hiroaki Kashima, Mr. Fumihiro Adachi and Mr. Ryosuke Ito for their cooperation in the work. I would like to thank all members of our laboratory for their helpful support and discussion.

Finally, I wish to thank my family and friends for their support.

Bibliography

[1] M. Araki, S. Nakagawa, J. Nomura, and S. Doshita. Incremental utterance understanding using phrase based plan recognition. In *Proc. of Int'l Conf. Computer Processing of Oriental Languages*, pages 353–356, 1999.

[2] H. Iida and H. Arita. Natural language understanding on a four-typed plan recognition model (in Japanese). *IPSJ Journal*, 31(6):810–821, 1990.

[3] K. Yamada, R. Mizoguchi, and N. Harada. User's utterance model and cooperative answering for question- answering systems (in Japanese). *IPSJ Journal*, 35(11):2265–2275, 1994.

[4] D. Sadek. Design considerations on dialogue systems: From theory to technology -the case of artimis-. In *Proc. ESCA workshop on Interactive Dialogue in Multi-Modal Systems*, 1999.

[5] Y. Niimi and Y. Kobayashi. A dialog control strategy based on the reliability of speech recognition. In *Proc. ICSLP*, 1996.

[6] S. Furui and K. Yamaguchi. Designing a multimodal dialogue system for information retrieval. In *Proc. ICSLP*, 1998.

[7] T. Watanabe, M. Araki, and S. Doshita. Evaluating dialogue strategies under communication errors using computer-to-computer simulation. *Trans. of IEICE, Info & Syst.*, E81-D(9):1025–1033, 1998.

[8] Y. Niimi, T. Nishimoto, and M. Araki. Relations between the efficiency of control strategies for confirmation dialogue and the performance of a speech recognizer (in Japanese). In *IPSJ Tech. Report*, 99-SLP-27-17, 1999.

[9] M. Araki, K. Komatani, T. Hirata, and S. Doshita. A dialogue library for task-oriented spoken dialogue systems. In *Proc. IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 1–7, 1999.

[10] DRI. Discourse research initiative, 1996. http://www.georgetown.edu/luperfoy/Discourse-Treebank/dri-home.html.

[11] J. Carletta, N. Dahlbäck, N. Reithinger, and M. Walker. Standards for dialogue coding in natural language processing. Report on the Dagstuhl-Senimar, 1997. http://www.dfki.uni-sb.de/dri/.

[12] A. Ichikawa, M. Araki, M. Ishizaki, S. Itabashi, T. Itoh, H. Kashioka, K. Kato, H. Kikuchi, T. Kumagai, A. Kurematsu, H. Koiso, M. Tamoto, S. Tutiya, S. Nakazato, Y. Horiuchi, K. Maekawa, Y. Yamashita, and T. Yoshimura. Standardising annotation schemes for japanese discourse. In *Proc. of the 1st International Conference on Language Resources and Evaluation (LREC'98)*, pages 731–736, 1998.

[13] A. Ichikawa, M. Araki, Y. Horiuchi, M. Ishizaki, S. Itabashi, T. Itoh, H. Kashioka, K. Kato, H. Kikuchi, H. Koiso, T. Kumagai, A. Kurematsu, K. Maekawa, S. Nakazato, M. Tamoto, S. Tutiya, Y. Yamashita, and T. Yoshimura. Evaluation of annotation schemes for japanese discourse. In *Proc. of ACL '99 Workshop on Towards Standards and Tools for Discourse Tagging (ACL-WS '99)*, pages 26–34, 1999.

[14] J.R. Searle. *Speech Acts.* Cambridge University Press, 1969.

[15] J. L. Austin. *How to Do Things with Words.* Oxford University Press, 1962.

[16] M. Kawamori and A. Shimazu. Analysis of utterance exchange in discourse (in Japanese). In *Tech. Report of IEICE*, NLC 95-73, pages 31–38, 1996.

[17] H. Koiso, Y. Horiuchi, S. Tutiya, and A. Ichikawa. The prediction of the termination / continuation of utterance based on some linguistic and prosodic elements (in Japanese). In *Proc. of the 10th Annual Conf. of JSAI*, pages 407–410, 1996.

[18] M. Meteer et al. Dysfluency annotation stylebook for the switchboard corpus. Distributed by LDC, 1995. ftp://ftp.cis.upenn.edu/pub/treebank/swbd/doc/DFL-book.ps.

[19] The TRAINS Project. The multiparty discourse group's dialog annotation scheme, 1996. http://www.cs.rochester.edu:80/research/trains/annotation.

[20] N. Takinaga, T. Nishimoto, and Y. Niimi. Automatically tagging of utterances in sightseeing guidance dialogs (in Japanese). In *JSAI Tech. Report*, SIG-SLUD-9801, pages 7–12, 1998.

[21] N. Reithinger and E. Maier. Utilizing stastical dialog act processing in verbmobil. In *Proc. of the 33rd Annual Meeting of the ACL*, pages 116–121, 1995.

[22] Y. Matsumoto, A. Kitauchi, T. Yamashita, O. Imaichi, and T. Imamura. Japanese morphological analysis system "ChaSen" version 1.0 document (in Japanese). NAIST Technical Report, NAIST-IS-TR97007, 1997.

[23] H.P. Grice. *Logic and conversation.* Harvard University Press, 1975.

[24] S. Itabashi. Simulated spoken dialogue corpus (CDROM), research on understanding and generating dialogue by integrated processing of speech, language and concept, vol. 1-4. Ministry of Education, Science and Culture, Japan, Grant-in-Aid for Scientific Research on Priority Areas., 1995. http://winnie.kuis.kyoto-u.ac.jp/taiwa-corpus/.

[25] S. Sutton, D. G. Novick, R. Cole, P. Vermeulen, J. de Villiers, J. Schalkwyk, and M. Fanty. Building 10,000 spoken dialogue systems. In *Proc. ICSLP*, 1996.

[26] S. Kaspar and A. Hoffmann. Semi-automated incremental prototyping of spoken dialog systems. In *Proc. ICSLP*, 1998.

[27] R. C. Moore. The challenge of domain-independent speech understanding. In *Proc. IEEE-ICASSP*, 1998.

[28] T. Kawahara, C.-H. Lee, and B.-H. Juang. Flexible speech understanding based on combined key-phrase detection and verification. *IEEE Trans. on Speech and Audio Processing*, 6(6):558–568, 1998.

[29] R. San-Segundo, B. Pellom, W. Ward, and J. Pardo. Confidence measures for dialogue management in the CU communicator system. In *Proc. IEEE-ICASSP*, 2000.

[30] L. F. Lamel, S. Rosset, J-L. S. Gauvain, and S. K. Bennacef. The LIMSI ARISE system for train travel information. In *Proc. IEEE-ICASSP*, 1999.

[31] J. Sturm, E. Os, and L. Boves. Issues in spoken dialogue systems: Experiences with the Dutch ARISE system. In *Proc. ESCA workshop on Interactive Dialogue in Multi-Modal Systems*, 1999.

[32] A. Lee, T. Kawahara, and S. Doshita. Large vocabulary continuous speech recognition parser based on A* search using grammar category-pair constraint (in Japanese). *Trans. IPSJ*, 40(4):1374–1382, 1999.

[33] O. Furuse, Y. Sobashima, T. Takezawa, and N. Uratani. Bilingual corpus for speech translation. In *Proc. AAAI'94 Workshop on the Integration of Natural Language and Speech Processing*, pages 84–91, 1994.

[34] T. Morimoto, N. Uratani, T. Takezawa, O. Furuse, Y. Sobashima, H. Iida, A. Nakamura, Y. Sagisaka, N. Higuchi, and Y. Yamazaki. Speech and language database for speech translation research. In *Proc. ICSLP*, pages 1791–1794, 1994.

[35] K. Tanaka, S. Hayamizu, Y. Yamasita, K. Shikano, S. Itahashi, and R. Oka. Design and data collection for a spoken dialogue database in the real world computing program. In *Proc. Acoustical Society of America and Acoustical Society of Japan Third Joint Meeting*, pages 1027–1030, 1996.

[36] D. J. Litman, M. A. Walker, and M. S. Kearns. Automatic detection of poor speech recognition at the dialogue level. In *Proc. of 37th Annual Meeting of the ACL*, 1999.

[37] C. Pao, P. Schmid, and J. Glass. Confidence scoring for speech understanding systems. In *Proc. ICSLP*, 1998.

[38] G. Bouwman, J. Sturm, and L. Boves. Incorporating confidence measures in the Dutch train timetable information system developed in the ARISE project. In *Proc. IEEE-ICASSP*, 1999.

[39] D. Goddeau, H. Meng, J. Polifroni, S. Seneff, and S. Busayapongchai. A form-based dialogue manager for spoken language applications. In *Proc. ICSLP*, 1996.

[40] S. Bennacef, L. Devillers, S. Rosset, and L. Lamel. Dialog in the RAILTEL telephone-based system. In *Proc. ICSLP*, 1996.

[41] T. Kawahara, K. Tanaka, and S. Doshita. Domain-independent platform of spoken dialogue interfaces for information query. In *Proc. ESCA workshop on Interactive Dialogue in Multi-Modal Systems*, pages 69–72, 1999.

[42] T. Kawahara, A. Lee, T. Kobayashi, K. Takeda, N. Minematsu, K. Itou, A. Ito, M. Yamamoto, A. Yamada, T. Utsuro, and K. Shikano. Japanese dictation toolkit – 1997 version –. *J. Acoust. Soc. Japan (E)*, 20(3):233–239, 1999.

[43] B. A. Hockey, D. Rossen-Knill, B. Spejewski, M. Stone, and S. Isard. Can you predict responses to yes/no questions? yes,no,and stuff. In *Proc. EUROSPEECH*, 1997.

[44] R. Pieraccini, E. Levin, and C-H. Lee. Stochastic representation of conceptual structure in the atis task. In *Proc. 4th Joint DARPA Speech and Natural Language Workshop*, 1991.

[45] A. Potamianos, E. Ammicht, and H.-K. J. Kuo. Dialogue management in the Bell labs communicator system. In *Proc. ICSLP*, 2000.

[46] A. Potamianos and J. Kuo. Statistical recursive finite state machine parsing for speech understanding. In *Proc. ICSLP*, volume 3, pages 510–513, 2000.

[47] E. Levin, S. Narayanan, R. Pieraccini, K. Biatov, E. Bocchieri, G. Di Fabbrizio, W. Eckert, S. Lee, A. Pokrovsky, M. Rahim, P. Ruscitti, and M. Walker. The AT&T-DARPA communicator mixed-initiative spoken dialogue system. In *Proc. ICSLP*, 2000.

[48] J. F. Allen, B. W. Miller, E. K. Ringger, and T. Sikorski. A robust system for natural spoken dialogue. In *Proc. of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, pages 62–70, 1996.

[49] J. Chu-Carroll and B. Carpenter. Dialogue management in vector-based call routing. In *Proc. of COLING-ACL98*, pages 256–262, 1998.

[50] E. Levin, R. Pieraccini, and W. Eckert. Learning dialogue strategies within the markov decision process framework. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 72–79, 1997.

[51] D. J. Litman, M. S. Kearns, S. Singh, and M. A. Walker. Automatic optimization of dialogue management. In *Proc. COLING*, pages 502–508, 2000.

[52] M. Denecke. An information-based approach for guiding multi-modal human-computer-interaction. In *Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, 1997.

[53] F. Adachi, K. Komatani, and T. Kawahara. Analysis and handling of unexpected utterances in a spoken dialogue system for information retrieval (in Japanese). In *JSAI Tech. Report*, SIG-SLUD-A001-2, 2000.

[54] http://www.sharp.co.jp/liquiy/liquiy02.html.

[55] K. Nagatomo, R. Nisimura, K. Komatsu, Y. Kuroda, A. Lee, H. Saruwatari, and K. Shikano. Complemental backoff algorithm for merging language models (in Japanese). In *IPSJ Tech. Report*, 2001-SLP-35-9, 2001.

[56] T. Kawahara, T. Sumiyoshi, A. Lee, K. Takeda, M. Mimura, A. Ito, K. Itou, and K. Shikano. Product software of continuous speech recognition consortium –2000 version– (in Japanese). In *IPSJ Tech. Report*, 2001-SLP-38-6, 2001.

[57] K. Komatani and T. Kawahara. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. COLING*, pages 467–473, 2000.

[58] K. Komatani, K. Tanaka, H. Kashima, and T. Kawahara. Domain-independent spoken dialogue platform using key-phrase spotting based on combined language model. In *Proc. EUROSPEECH*, pages 1319–1322, 2001.

[59] A. Lee, T. Kawahara, K. Takeda, and K. Shikano. A new phonetic tied-mixture model for efficient decoding. In *Proc. IEEE-ICASSP*, pages 1269–1272, 2000.

List of Publications by the Author

Major Publications

[1] K. Komatani, M. Araki, and S. Doshita. A method to infer the utterance-unit tag in dialogue corpus (in Japanese). *Journal of JSAI*, Vol. 14, No. 2, pp. 273–281, 1999.

[2] M. Araki, K. Komatani, T. Hirata, and S. Doshita. A dialogue library for task-oriented spoken dialogue systems. In *Proc. IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pp. 1–7, 1999.

[3] K. Komatani and T. Kawahara. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. Int'l Conf. Computational Linguistics (COLING)*, pp. 467–473, 2000.

[4] T. Kawahara, K. Komatani, and S. Doshita. Dialogue management using concept-level confidence measures of speech recognition. In *Proc. Int'l Sympo. on Spoken Dialogue*, 2000.

[5] K. Komatani and T. Kawahara. Generating effective confirmation and guidance using two-level confidence measures for dialogue systems. In *Proc. Int'l Conf. on Spoken Language Processing (ICSLP)*, Vol. 2, pp. 648–651, 2000.

[6] K. Komatani, K. Tanaka, H. Kashima, and T. Kawahara. Domain-independent spoken dialogue platform using key-phrase spotting based on combined language model. In *Proc. European Conf. Speech Commun. & Tech. (EUROSPEECH)*, pp. 1319–1322, 2001.

[7] R. Ito, K. Komatani, and T. Kawahara. Spoken dialogue help system for electrical appliances using knowledge and structure of their manuals (in Japanese). *IPSJ Journal*, Vol. 43, No. 7, pp. 2147–2154, 2002.

[8] K. Komatani, T. Kawahara, R. Ito, and H.G. Okuno. Efficient dialogue strategy to find users' intended items from information query results. In *Proc. Int'l Conf. Computational Linguistics (COLING)*, Vol. 1, pp. 481–487, 2002.

[9] K. Komatani and T. Kawahara. Flexible dialogue management for generating efficient confirmation and guidance using confidence measures of speech recognition result (in Japanese). *IPSJ Journal*, Vol. 43, No. 10, pp. 3078–3086, 2002.

[10] K. Komatani, H. Kashima, K. Tanaka, and T. Kawahara. Domain-independent spoken dialogue platform for database query using key-phrase spotting based on combined language model (in Japanese). *IPSJ Journal*, (submitted for review).

Technical Reports

[1] K. Komatani, M. Araki, and S. Doshita. Inference of illocutionary act tag and its implementation to gui tool to support discourse tagging (in Japanese). In *JSAI Tech. Report*, SIG-SLUD-9801-5, 1998.

[2] M. Araki, K. Komatani, T. Hirata, and S. Doshita. A dialogue library for task-oriented spoken dialogue systems (in Japanese). In *JSAI Tech. Report*, SIG-SLUD-9901-1, 1999.

[3] K. Komatani and T. Kawahara. A robust mixed-initiative dialogue system using confidence measures of speech recognition results (in Japanese). In *IPSJ Tech. Report*, SLP-30-9, 2000.

[4] F. Adachi, K. Komatani, and T. Kawahara. Analysis and handling of unexpected utterances in a spoken dialogue system for information retrieval (in Japanese). In *JSAI Tech. Report*, SIG-SLUD-A001-2, 2000.

[5] R. Ito, K. Komatani, and T. Kawahara. Spoken dialogue help system for electrical appliances using knowledge and structure of their manuals (in Japanese). In *IPSJ Tech. Report*, SLP-37-1, 2001.

[6] K. Komatani, T. Kawahara, Y. Kiyota, S. Kurohashi, and P. Fung. Resraurant search system with speech interface using flexible language model and matching (in Japanese). In *Tech. Report of IEICE*, SP2001-113, NLC2001-78 (SLP-39-30), 2001.

[7] S. Ueno, K. Komatani, T. Kawahara, and H.G. Okuno. Generation of cooperative responses using user model in spoken dialogue system (in Japanese). In *IPSJ Tech. Report*, SLP-42-2, 2002.

Oral Presentations

[1] K. Komatani, M. Araki, and S. Doshita. Inference of illocutionary-act tags using surface information of sentences in dialogue corpora (in Japanese). In *Proc. Meeting of the Association for NLP*, pp. 406–409, 1998.

[2] K. Komatani and T. Kawahara. Dialogue management for coping with speech recognition error in a mixed-initiative dialogue (in Japanese). In *Proc. Meeting of the Association for NLP*, pp. 336–339, 2000.

[3] K. Komatani and T. Kawahara. Using confidence measures of speech recognition results in spoken dialogue system (in Japanese). In *Proc. Meeting Acoust. Soc. Japan*, 3-5-2, fall 2000.

[4] K. Komatani, H. Kashima, F. Adachi, and T. Kawahara. A portability-oriented spoken dialogue platform for database query (in Japanese). In *Proc. Meeting of the Association for NLP*, pp. 113–116, 2001.

[5] K. Komatani, R. Ito, and T. Kawahara. Dialogue strategy to narrow down the search results in information search systems (in Japanese). In *Proc. Meeting of the Association for NLP*, pp. 252–255, 2002.

[6] K. Komatani, S. Ueno, T. Kawahara, and H.G. Okuno. Generation of cooperative responses using user model in spoken dialogue system (in Japanese). In *Information Technology Letters (FIT2002)*, Vol. 1, pp. 95–96, 2002.